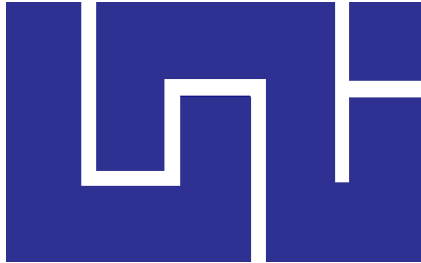


**UNIVERSIDAD NACIONAL DE INGENIERÍA**

**Facultad de Electrotecnia y Computación**



**Propuesta para Implementación de generador de reportes parametrizados  
para el Banco de Proyectos del Sistema Nacional de Inversión Pública**

Trabajo de Diploma  
Para optar al título de ingeniero en computación

Presentado por:

*Br. Javier Alexander Acevedo Rostrán*

*Br. Paul Antonio Zamora Rodríguez*

Tutor: Ing. José L. Díaz Chow.

Abril de 2016

Managua, Nicaragua



*Propuesta para Implementación de generador de reportes parametrizados para  
el Banco de Proyectos del Sistema Nacional de Inversión Pública.*

## **Dedicatoria**

### **Javier Alexander Acevedo Rostrán**

- ❖ Dedico este logro a esos seres especiales que han estado ahí con nosotros en nuestras altas y bajas, brindándonos su apoyo incondicional y conduciéndonos por el trayecto de la vida.

### **Paul Antonio Zamora Rodríguez**

- ❖ Primeramente agradecer a DIOS todopoderoso por las bendiciones que nos regala cada día, el tiempo que nos regala para realizar cada una de sus pruebas y la sabiduría que permitió culminar esta meta.
- ❖ Segundo a la razón de seguir adelante y mi inspiración mi hija Allison Zamora que gracias a ella me da fuerzas para seguir adelante con su sonrisa y cada momento que comparto con ella.
- ❖ A mis padres Ing. Rony Zamora e Ing. Sofía Rodríguez que gracias a su esfuerzo y ejemplo he podido crecer a nivel profesional, y cada día sus enseñanzas han inculcado ese deseo de ser mejor, cada momento de mi vida.
- ❖ A mis compañeros y amigos que me han dado su apoyo incondicional para crecer como persona porque cada risa y cada momento difícil supimos sobrellevarlo de la mejor forma.

## **Agradecimiento**

- ❖ Agradecer a Dios por cada suspiro que nos permite seguir adelante, el tiempo efímero que nos regala en este mundo para cumplir las metas propuestas y el agrado de disfrutar las experiencias buenas y malas que nos hacen crecer a nivel personal.
- ❖ A nuestros familiares por el apoyo incondicional que nos brindaron tanto moral como económico que nos permitieron seguir adelante en esta meta propuesta.
- ❖ Al ingeniero José Díaz Chow por su nata labor como profesor y tutor, por ser una persona incondicional al momento de ser nuestro tutor, a la minuciosa labor de revisar cada detalle de este esfuerzo realizado en conjunto ya que nos ha enseñado el valor del esfuerzo y ha ayudado en gran manera al crecimiento profesional de cada uno de sus alumnos y en especial a nosotros.
- ❖ Finalmente a cada uno de los profesores y autoridades de la UNI por el tiempo y la enseñanza compartida a lo largo de estos 6 años de gran esfuerzo. Sin olvidar también a ese millar de palabra escritas en folletos, libros de gran inspiración que nos ayudaron en el camino del entendimiento para lograr esta meta.

## **Resumen**

El presente trabajo monográfico “Propuesta para Implementación de generador de reportes parametrizados para el Banco de Proyectos del Sistema Nacional de Inversión Pública” tiene como propósito diseñar e implementar, desde sus cimientos, un nuevo generador de reportes parametrizados para el Banco de Proyectos, basado en tecnología actualizada y patrones de diseño que permitan facilitar y agilizar el servicio de salida de información en reportes tabulares, mejorando el tiempo de respuesta del sistema respecto al estado actual.

La realización de este trabajo se ha organizado en dos partes. La primera parte consistente en el desarrollo del generador de reportes, basado en metodologías, procedimientos, recomendaciones de mejores prácticas, técnicas y estándares de la ingeniería de software; y la segunda en la evaluación de los resultados, empleando las técnicas y procedimientos de evaluación de sistemas.

La primera fase se organizó siguiendo la metodología estándar de desarrollo de software RUP (Rational Unified Process), el cual aborda el desarrollo del proyecto mediante un proceso iterativo incremental, basado en cuatro fases: inicio o conceptualización, elaboración, construcción y transición. El lenguaje para el desarrollo de la aplicación fue Visual C# y el gestor de base de datos utilizado fue Microsoft SQL Server 2008 R2.

En la segunda fase se realizó la evaluación de la solución, obteniéndose que tanto el proceso de diseño de reportes como el de generación de los mismos es más eficiente con el nuevo generador de reportes, alcanzándose un 45% de mejora de rendimiento en el diseño y hasta un 77% en la generación de los reportes.

# **Contenido**

Dedicatoria.....	i
Agradecimiento.....	ii
Resumen .....	iii
1    Introducción.....	1
1.1    Antecedentes .....	2
2    Objetivos .....	5
2.1    Objetivo General .....	5
2.2    Objetivos Específicos .....	5
3    Justificación.....	6
4    Marco Teórico.....	8
4.1    SQL Server .....	9
4.1.1    SQL Server 2008 R2 y SQL Server Management Studio .....	10
4.1.2    Diferencias notables entre las versiones de SQL Server 2005 y 2008 .....	12
4.2    Microsoft Visual Studio 2010 .....	13
4.3    Syncfusion Essential Studio 2015.....	15
4.4    Microsoft Entity Framework .....	18
4.5    Bootstrap .....	20
4.6    JQuery .....	22
4.6.1    Metodología de Desarrollo de Software.....	23
5    Desarrollo del Software Generador de Reportes .....	26
5.1    Fase de Conceptualización.....	26
5.1.1    Requerimientos Generales .....	26
5.1.2    Reglas del negocio.....	29
5.1.3    Requerimientos Específicos por Aplicativo .....	29
5.1.4    Estudio de Factibilidad .....	31
5.2    Fase de Elaboración .....	42
5.2.1    Modelado de Casos de Uso.....	42

5.2.2	Casos de Uso para la Herramienta Diseñadora de Reportes .....	43
5.2.3	Casos de Uso del Servidor de Reportes .....	48
5.2.4	Modelado conceptual o de Dominio .....	50
5.2.5	Diagrama de Clases .....	53
5.2.6	Diseño de base de datos .....	54
5.2.7	Diagrama de Actividades .....	56
5.2.8	Diagrama de Secuencia .....	57
5.2.9	Arquitectura Ejecutable .....	65
5.3	Fase de Construcción .....	68
5.3.1	Construcción de la Herramienta Diseñadora de Reportes .....	68
5.3.2	Construcción del Servidor de Reportes.....	71
5.3.3	Construcción de la Máquina Generadora de Reportes.....	74
5.4	Fase de Transición.....	76
5.4.1	Arquitectura de Despliegue .....	76
5.4.2	Proceso de Instalación .....	79
5.4.3	Inducción a usuarios .....	83
5.4.4	Pruebas de usuario .....	83
5.4.5	Documentación.....	84
6	Evaluación de resultados .....	85
6.1	Métricas .....	85
6.2	Método de evaluación .....	87
6.2.1	Procedimiento para evaluación de la creación de un reporte.....	88
6.2.2	Procedimiento para evaluación de la generación de un reporte .....	88
6.3	Resultados de la evaluación del proceso de creación de reportes.....	89
6.3.1	Evaluación de Desempeño.....	89
6.3.2	Evaluación de Facilidad de uso .....	90
6.4	Evaluación del proceso de generación de reportes.....	91
6.4.1	Evaluación del Desempeño.....	91
6.4.2	Evaluación de Facilidad de uso. ....	97
7	Conclusiones y Recomendaciones .....	99
7.1	Conclusiones .....	99
7.2	Recomendaciones.....	101



8	Bibliografía .....	102
9	Anexos.....	104
9.1	ANEXO1. Formato o ficha de registro de resultados. ....	104
9.1.1	Ficha de registro de resultados Creación de Reportes. ....	104
9.1.2	Ficha de registro de resultados Generación de Reportes.....	105
9.2	ANEXO2. Guía de Referencia Técnica del Generador de Reportes. ....	106
9.2.1	Aspectos Generales.....	106
9.2.2	Configuración .....	109
9.2.3	Funciones .....	111
9.2.4	Procedimientos Almacenados .....	111
9.3	ANEXO 3. Manual de Usuario de la Herramienta Diseñadora de Reportes. ....	114
9.4	ANEXO 4. Manual de Usuario del Servidor de Reportes.....	125
9.5	ANEXO 5. Procedimiento para exportación de la Base de Datos. ....	130

# **1 Introducción**

El Sistema Nacional de Inversión Pública (SNIP) es un sistema administrativo del estado que rige la inversión pública del país con el fin que responda a las estrategias y políticas de crecimiento y desarrollo socioeconómico de la nación. Para el logro este propósito el sistema se auxilia en una serie de instrumentos. Uno de sus principales instrumentos es el Banco de Proyectos (BP), que es un sistema de gestión de información basado en la Web, que constituye el registro oficial de las iniciativas de inversión pública del país, sistematiza el proceso de inversión, consolida la información de la inversión pública nacional y apoya a la toma de decisiones.

La principal salida de información del Banco de Proyectos es en forma de reportes tabulares a hojas de cálculo de Excel. Actualmente la generación de reportes es muy ineficiente debido una serie de factores como la obsolescencia de la tecnología empleada y limitaciones de diseño ante el volumen actual de datos, que ralentizan el proceso de generación, llegando incluso al *time-out* del navegador en algunos casos, afectando sensiblemente la productividad de los usuarios.

El presente trabajo consistió en el diseño e implementación de un nuevo generador de reportes parametrizados con salida a formato de Excel, para el Banco de proyectos, el cual está basado en las últimas versiones y perfeccionamientos en tecnología de la plataforma que lo soporta, y posee un diseño más amigable al usuario que permite facilitar su uso y mejorar el rendimiento de la ejecución de los reportes, aliviando la problemática presentada.

A continuación se presentan en detalle los antecedentes que dieron origen y definieron la a la necesidad de este trabajo, y posteriormente, la metodología empleada para su diseño y ejecución, lo que permitió alcanzar los resultados propuestos.

## **1.1 Antecedentes**

El Banco de Proyectos es un sistema de información con una larga trayectoria de evolución funcional y tecnológica. Se ha venido desarrollando desde sus inicios en 1996, como aplicativo de escritorio en plataforma MS-DOS codificado en lenguaje de programación Clipper<sup>1</sup> con base de datos basada en archivos libres con estructura DBASE, pasando por una migración a plataforma Windows de 32 bits en 1999, desarrollado en Visual Basic 6.0 y base de datos MS-Jet, hasta la versión actual consistente en un sistema de clase empresarial basado en web, desarrollado sobre la plataforma .Net con base de datos MS-SQL Server a partir de 2003 hasta la fecha.

En su versión web, el sistema se ha organizado de forma modular de acuerdo a los subsistemas que conforman las fases del ciclo de gestión de la inversión pública: pre-inversión, programación y seguimiento a la ejecución. Cada subsistema tiene uno o más módulos específicos de acuerdo a las tareas que realizan los usuarios en su operación.

Estos módulos se han ido desarrollando a lo largo de doce años, empleando diferentes versiones de entornos y marcos de trabajo, lenguajes, estilos de programación y de presentación de la interfaz de usuario. Esta falta de estandarización ha sido causada principalmente por los cambios en la tecnología al momento del desarrollo de cada módulo y el personal a cargo del mismo.

El módulo de reportes actual fue implementado en 2003 sobre el Framework .Net 1.1. Su codificación se realizó en lenguaje Visual Basic .Net. Los principales requerimientos técnicos de este módulo de reportes, además de ser basado en web, fue que generara reportes tabulares con salida a Excel.

---

<sup>1</sup> Clipper es un lenguaje de programación procedural creado en 1985 por Nantucket Corporation y vendido posteriormente a Computer Associates, desarrollado para facilitar la generación de programas ejecutables que interactuaran con bases de datos dBase III.

El hecho que los reportes tuvieran que generarse a Excel, representó un reto grande para el equipo de desarrollo involucrado con la tarea en ese entonces, debido a que en ese momento ningún componente de generación de reportes para .Net brindaba esa funcionalidad. Se decidió enfrentar el reto de forma similar como se había hecho en la versión desktop para Windows: generar el reporte a texto e importarlo a Excel, formateándolo mediante macros en VBA. El problema de este enfoque es que requiere interoperabilidad con COM, lo que representa un riesgo grande para el servidor web, por lo cual se decidió dividir el proceso en dos partes: generar el reporte a texto en el servidor web, y la parte de importación a Excel hacerlo en la máquina cliente, para lo cual el usuario debe descargar e instalar un pequeño aplicativo y una plantilla base conteniendo las macros. Esta forma de trabajo ha sido inconveniente en algunos casos donde el usuario empresarial no tiene permiso de instalación y más aún con los mecanismos de seguridad de los navegadores modernos que requieren que los programas descargados del internet estén firmados digitalmente para considerarlos seguros.

Actualmente, las diferencias en los procesos de desarrollo de los módulos han generado incompatibilidades a nivel de plataforma de desarrollo, marco de trabajo y uso de normativas y estándares, que dificulta el nivel de integración requerido entre los sistemas, llevando a una pérdida de desempeño en su operación.

Para resolver esta problemática, la Dirección General de Inversión Pública (DGIP) del Ministerio de Hacienda y Crédito Público (MHCP), ente rector del Sistema Nacional de Inversión Pública, que tiene a su cargo los procesos de implementación del software del Banco de Proyectos, decidió llevar a cabo un proceso de reingeniería de los módulos a fin que compartan un marco de trabajo, entorno de desarrollo y estilo de presentación comunes.

Uno de los componentes clave de este proceso de reingeniería es el desarrollo de un módulo de reportes totalmente nuevo, dado que el actual además de la problemática de incompatibilidad planteada, presenta recientemente un problema

de pobre desempeño en la generación de los reportes, que está afectando seriamente la productividad de los usuarios.

El generador de reportes del Banco de Proyectos requerido debe permitir a los usuarios generar diferentes reportes tabulares con salida a hojas de cálculo de Excel. Estos reportes deben ser paramétricos, es decir, la naturaleza y organización de la información solicitada se puede especificar en función de parámetros que el usuario selecciona. Uno de las capacidades paramétricas más poderosa es la selección de grupos o niveles de agrupación, mediante la cual un mismo reporte puede generar múltiples salidas con diferentes esquemas de ordenación de los mismos datos.

El presente trabajo pretende proponer una solución que, a partir de los requerimientos expresados para el nuevo generador de reportes, y sobre la base de un diseño apropiado, empleando las versiones y perfeccionamientos de tecnología más recientes de la plataforma del Banco de Proyectos, permita superar la problemática y limitaciones tecnológicas del módulo de reportes actual.

## **2 Objetivos**

### **2.1 Objetivo General**

Proveer al Banco de Proyectos del Sistema Nacional de Inversión Pública un nuevo módulo de generación de reportes tabulares parametrizados, basado en tecnología actualizada y un diseño mejorado, que permita facilitar y agilizar la creación y generación de reportes.

### **2.2 Objetivos Específicos**

- Analizar los requerimientos técnicos necesarios para la implementación de un generador de reportes de acuerdo a las necesidades de los usuarios del Banco de Proyectos y las normativas definidas por la Dirección General de Inversión Pública para la reingeniería del Banco de Proyectos.
- Elaborar el diseño del nuevo generador de reportes del Banco de Proyectos a partir de los requerimientos y normativas definidas por la Dirección General de Inversión Pública para la reingeniería del Banco de Proyectos.
- Implementar la aplicación para elaboración y publicación de reportes sobre la base del diseño realizado.
- Implementar los componentes de la máquina de generación de reportes sobre la base del diseño realizado.
- Implantar el nuevo módulo generador de reportes en el Banco de Proyectos y evaluar si cumple con los requerimientos de facilidad de uso y desempeño esperados.

### 3 Justificación

La Dirección de Desarrollo del Banco de Proyectos (DDBP), oficina de la Dirección General de Inversión Pública (DGIP) encargada de la conceptualización, desarrollo y administración del Banco de Proyectos<sup>2</sup>, a partir de los reportes de incidentes de los usuarios y pruebas de evaluación del sistema determinó que la generación de reportes tiene actualmente una latencia<sup>3</sup> muy alta, llegando incluso al *time-out*<sup>4</sup> del navegador en algunos casos, con lo que el usuario no obtiene su reporte de parte del sistema, y tiene que solicitar asistencia técnica, o generar otros reportes para obtener la información deseada mediante procesamiento posterior. Dada la alta demanda del servicio de generación de reportes, esta pérdida de desempeño se ha convertido en un problema que está afectando en gran medida la productividad de los usuarios del Banco de Proyectos, y por ende, la eficiencia de la gestión de la DGIP en su conjunto.

Las evaluación realizada por la DDBP<sup>5</sup> al módulo generador de reportes del Banco de Proyectos, determinó que las causas principales de la latencia exhibida por el módulo son: a) la obsolescencia de la tecnología empleada en su desarrollo (.Net Framework<sup>6</sup> versión 1,1, lanzada hace más de doce años y que es incompatible con las nuevas versiones del Framework a partir de la versión 2.0) y b) las limitaciones del diseño original ante el alto volumen actual de datos que se manejan.

---

<sup>2</sup> Función establecida en el Manual de Funciones de la DGIP. Ver referencia: (DGIP, 2011)

<sup>3</sup> Latencia: Medida de desempeño que representa el tiempo de respuesta de un sistema (Hennessy & Patterson, 2012).

<sup>4</sup> Los navegadores de Internet tienen definido un tiempo de espera máximo de 30 segundos para recibir respuesta del servidor web. Cuando la espera supera ese umbral (conocido como *time-out* por su denominación en inglés), el navegador da por perdida la conexión y descarta la solicitud, cancelándose el proceso que se estuviera llevando a cabo. Ver referencia: (Wikipedia, 2015)

<sup>5</sup> Información tomada de la sección de conclusiones del reporte interno de evaluación al módulo de generación de reportes parametrizados desarrollado por la DDBP. Ver referencia: (DDBP-DGIP, 2013)

<sup>6</sup> El .Net Framework es la infraestructura tecnológica de la empresa Microsoft para el desarrollo de software para sistemas Windows.

Adicionalmente, en el generador actual se requiere descargar e instalar un aplicativo y una plantilla en la máquina del cliente para realizar el formateo a Excel, lo que ha generado problemas derivados de permisos de instalación y denegación de descarga por los mecanismos de seguridad de los navegadores modernos, reduciendo sensiblemente el uso del generador en las instituciones cliente del Banco de Proyectos.

Como vía de solución a esta problemática, el presente trabajo propone diseñar e implementar un nuevo generador de reportes parametrizados para el Banco de Proyectos, basado en tecnología actualizada y un diseño mejorado que permita facilitar y agilizar este servicio, mejorando el tiempo de respuesta del sistema respecto al estado actual.

Al implementar esta propuesta se obtendrán los siguientes beneficios respecto al módulo de generación anterior:

- No habrá incompatibilidad tecnológica del módulo generador con otros módulos, lo que facilitará su integración en el Banco de Proyectos.
- El procedimiento de selección de parámetros, generación y descarga del reporte al usuario final será más fácil para el usuario.
- No habrá necesidad de instalar ninguna aplicación en la máquina del cliente porque el reporte se genera directamente en formato de Excel.
- Se habrá mejorado el desempeño del proceso de generación de reportes al reducirse el tiempo requerido para obtener los reportes.
- Todos los reportes disponibles se generarán ágilmente, si llegar a *time-outs* del navegador, contribuyendo así a una mejor eficiencia y productividad de los usuarios.
- El procedimiento para diseñar y publicar reportes será más rápido y simplificado, con lo que se mejorará la eficiencia de esta actividad.
- Se facilitará el uso y mejorará el rendimiento del módulo con lo que se incrementará la satisfacción del usuario final.



- Se mejorará el modelo de datos del generador de reportes, haciéndolo independiente del sistema, por lo cual puede ser fácilmente aplicable en otros sistemas y potencialmente, un producto open source o comercial.

## **4 Marco Teórico**

La tecnología se ha ido desarrollando a gran media con el pasar de los últimos años. Hemos transitado un largo camino de mejoramiento de poder y reducción de tamaño desde la primera computadora Eniac en 1945 que cubría el área de un campo de fútbol, hasta las computadoras actuales de alta gama y procesamiento de información que caben en un equipo de escritorio, como por ejemplo computadoras con procesadores de Intel Core i7 que soportan software de última tecnología como el sistema operativo Windows 10 y el gestor de base de datos SQL Server 2014.

Según cita Corrales en (Corrales, 2014), podemos definir el desarrollo tecnológico como “la Intensificación del empleo de la tecnología para elevar el nivel económico de una región o para proporcionar medios concretos que mejoren el rendimiento de una función o programa de producción”. En este sentido, los sistemas de información son herramientas que aportan al desarrollo tecnológico, que a medida que se van consolidando e integrando nuevas innovaciones tecnológicas van ayudando en mayor medida a las organizaciones para registrar y clasificar de forma ordenada grandes volúmenes de datos en forma masiva.

Esto lo podemos apreciar en el proceso de evolución del Banco de Proyectos del SNIP, sistema de información oficial de la inversión pública del gobierno de Nicaragua, que tiene una larga trayectoria de desarrollo y perfeccionamiento desde sus inicios en 1996, donde comenzó como aplicación de escritorio para sistema operativo MS-DOS ejecutado en la consola de Windows 3.1, desarrollado en el lenguaje de programación Clipper con base de datos en archivos de dBase III, pasando luego a ser una aplicación de escritorio para Windows, con arquitectura de 32 bits, desarrollada sobre la plataforma COM empleando Visual

Studio Enterprise versión 6.0, con lenguaje Visual Basic y Microsoft Jet como base de datos, que permitía acceso compartido en red; para finalmente establecerse como un sistema de clase empresarial centralizado, sobre plataforma web, operado en línea a través del Internet por todas las instituciones del Estado que realizan inversión pública. Este sistema está desarrollado sobre la plataforma tecnológica de Microsoft .Net, empleando como gestor de base de datos Microsoft SQL Server estándar.



Ilustración 1: Logo de SQL Server estándar

#### **4.1 SQL Server**

Es un sistema de gestión de bases de datos relacionales (RDBMS) de la empresa de software Microsoft, que está diseñado para el entorno empresarial (Microsoft, inc., 2016). La interacción con SQL Server se ejecuta empleando el lenguaje de gestión de datos Transact – SQL (T-SQL), un conjunto de extensiones de programación de Sybase y Microsoft que añaden varias características al lenguaje SQL Estándar, incluyendo control de transacciones, excepción y manejo de errores, procesamiento de fila, así como variables declaradas. Gartner (Gartner Corporation, 2015) en su último reporte, sitúa a SQL server como uno de los cuatro líderes del mercado, segundo en posición, de bases de datos en el mundo, compitiendo con Oracle, IBM y SAP en el extremo superior derecho de su cuadrante mágico.

El código original de SQL Server fue desarrollado por Sybase a finales de 1980 con el nombre comercial de SyBase<sup>7</sup>. Posteriormente, Microsoft, Sybase y Ashton-Tate colaboraron para producir la primera versión del producto con el nombre SQL Server para sistema operativo OS/2. Sybase y Microsoft emprendieron iniciativas separadas, Sybase cambió el nombre de su producto a Adaptive Server Enterprise y Microsoft continuó el desarrollo de SQL Server.

Los cambios tecnológicos, como se mencionó anteriormente, afectan no solamente la forma en que se manejan los datos, sino también a grandes compañías que hacen mejoras y crean el desarrollo de los mismos programas que ayudan a demás instituciones tal es el caso de SQL Server que ha cambiado conforme al pasar de los años nombrando así sus versiones.

Para fines de desarrollo de la monografía fue requerido emplear una versión compatible con la plataforma de producción de los sistema de información de la DGIP (MS SQL Server 2005), por eso se empleó la versión MS SQL Server 2008 R2 Enterprise para el manejo de los datos.



Ilustración 2: Logo de MS SQL Server 2008 R2

#### **4.1.1 SQL Server 2008 R2 y SQL Server Management Studio**

---

<sup>7</sup> Ver mayores detalles en referencia: (Lambert, 2011)

Es un sistema de bases de datos que permite crear soluciones organizacionales que son más fáciles planificar, implementar y administrar de que las versiones anteriores de SQL Server.

SQL Server Management Studio es un entorno integrado para obtener acceso, configurar, administrar y desarrollar todos los componentes de SQL Server. SQL Server Management Studio combina un amplio grupo de herramientas gráficas con una serie de editores de script enriquecidos que permiten a desarrolladores y administradores de todos los niveles obtener acceso SQL Server.

SQL Server Management Studio combina las características del Administrador corporativo, el Analizador de consultas y Analysis Manager, herramientas incluidas en versiones anteriores de SQL Server, en un único entorno. Además, SQL Server Management Studio funciona con todos los componentes de SQL Server, como Reporting Services e Integration Services. De este modo, los desarrolladores pueden disfrutar de una experiencia familiar y los administradores de bases de datos disponen de una herramienta única y completa que combina herramientas gráficas fáciles de usar con funciones avanzadas de scripting.

Entre sus características SQL Server 2008 posee<sup>8</sup>:

- Seguridad: permite proteger columnas, tablas, base de datos completas y archivos donde se almacena la información; si alguien se lleva el archivo no podrá acceder a la información sin la clave de protección.
- Mejoras en el espejo de información (Mirroring). Se le conoce espejo o mirror a una copia directa de los datos colocada en otra ubicación
- Reparación automática de fallas.
- SQL 2008 recupera entre 823 y 824 errores sucedidos en el disco solicitando una copia reciente de los datos dañados al otro equipo espejo de manera transparente.
- Mejoras en el rendimiento de espejo de información.

---

<sup>8</sup> Ver referencia: (MacroH, 2012)

- SQL 2008 comprime la información de su bitácora (log) para reducir el ancho de banda requerido para realizar el proceso de espejo de información.
- Ahora con el nuevo modelo del .NET Framework 3.5, es posible realizar las consultas directamente desde el código de .NET, sin la necesidad de utilizar objetos de acceso a SQL como Comandos, Conexiones, Data Reader etc. El modelo de LINQ permite ejecutar consultas directas en tablas y campos de SQL desde mi código C#
- Mejoras en Datos y tipos de datos
- Parámetros de tipo Tabla. SQL 2008 permite pasar como parámetros a Stored Procedures y Funciones valores de tipo tabla, lo que permitirá segmentar, reutilizar y mejorar la lógica de mucha programación TSQL

#### **4.1.2 Diferencias notables entre las versiones de SQL Server 2005 y 2008**

<b>SQL 2005</b>	<b>SQL 2008</b>
1 No poder habilitar el encriptamiento de toda la base de datos	A. Se puede encriptar toda la base de datos
2. No provee respaldo de encriptación	B. Posee respaldo de encriptación
3. El tipo de datos XML es presentado	C. El tipo de dato XML es implementado y usado
4. No se encuentra Filestream	D. Filestream es presentado
5. No posee LINQ	E. Linq es presentado para recuperar múltiples tipos de datos
6. Los parámetros de los valores de tabla no están en SQL 2005	F. Se presentan los parámetros de los valores de tablas
7. La declaración merge no se encuentra en esta versión	G. La declaración merge es incluida en esta versión
8. Datetime es usado para ambos tiempo y fecha	H. Mayores cambios en Datetime, 4 tipos de datos son presentados, que son Date, time, Datetimeoffset and datetime 2

SQL 2005	SQL 2008
9. En SQL 2008, UDTs estaban restringidos a un máximo de 5 kb de capacidad	I. La restricción ha sido removida para UDTs en SQL 2008
10. No se presenta la sincronización de datos	J. La sincronización de datos es presentada

Tabla 1, SQL 2005 vs SQL 2008

Las características de SQL son integradas a un marco de trabajo que ayuda a la automatización de procesos y realizar interfaces que sean más agradable al usuario tanto en web como en aplicaciones de escritorio, mencionado esto, es necesario asimismo indicar que existe una herramienta que sobresale en el desarrollo de aplicaciones conectadas a SQL Server, el Microsoft Visual Studio el que incorpora una serie poderosa de compiladores, diseñadores, generadores de código y herramientas para el diseño, implementación y despliegue de aplicaciones sobre el marco de trabajo Microsoft .Net, sobre el que están desarrollados los aplicativos del Banco de Proyectos del SNIP, y por tanto, sobre este mismo marco de trabajo deberá desarrollarse la solución de generación de reportes.



Ilustración 3, Microsoft Visual Studio 2010

## 4.2 Microsoft Visual Studio 2010

Es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones de escritorio para Windows, aplicaciones Web ASP.NET, MVC y

Silverlight, Servicios Web XML, y aplicaciones móviles (Microsoft, Inc., 2016). Provee un único entorno integrado de desarrollo (IDE), que facilita la programación con soporte a diferentes lenguajes como Visual Basic, Visual C#, F# y Visual C++. Todas las aplicaciones desarrolladas en el Visual Studio tienen acceso a utilizar las funciones de .NET Framework, así como a herramientas de diseño y apoyo que permiten simplificar el desarrollo de aplicaciones y servicios Web.

Entre sus más destacables características, se encuentran la capacidad para utilizar múltiples monitores, así como la posibilidad de desacoplar las ventanas de su sitio original y acoplarlas en otros sitios de la interfaz de trabajo. Además ofrece la posibilidad de crear aplicaciones para muchas plataformas de Microsoft, como Windows, Azure, Windows Phone y Sharepoint. Microsoft ha sido sensible a la nueva tendencia de las pantallas táctiles y con este Visual Studio 2010 también es posible desarrollar aplicativos para pantallas multitáctiles.

Entre las versiones del Visual Studio se tienen las versiones gratuitas para desarrolladores:

- Visual Basic Express Edition
- Visual C# Express Edition
- Visual C++ Express Edition
- Visual J# Express Edition (Desapareció en Visual Studio 2008)
- Visual Web Developer Express Edition (para programar en ASP.NET)
- Visual F# Express Edition

Para el desarrollo de la aplicación se tenían a disposición diferentes tipos de ediciones de Visual Studio 2010 entre ellas se encuentran

**Visual Studio 2010 Premium:** Un conjunto de herramientas completo que simplifica el desarrollo de aplicaciones para personas o equipos que entregan

aplicaciones escalables de alta calidad, usando herramientas poderosas que funcionan de una manera simplificada e intuitiva.

**Visual Studio 2010 Professional:** La herramienta esencial para las personas que realizan tareas de desarrollo básico. Visual Studio 2010 Professional simplifica la compilación, la depuración y el despliegue de las aplicaciones en una variedad de plataformas incluyendo SharePoint y la Nube. También viene con el soporte integrado para el desarrollo con pruebas y con las herramientas de depuración que ayudan a garantizar unas soluciones de alta calidad.

**Visual Studio Team Foundation Server 2010:** Una plataforma de colaboración en el centro de la solución de gestión del ciclo de vida de una aplicación (ALM) de Microsoft. Team Foundation Server 2010 automatiza el proceso de entrega del software y le da las herramientas que necesita para gestionar eficazmente los proyectos de desarrollo de software a través del ciclo de vida de sistemas de tecnología de la información.

**Visual Studio Team Explorer Everywhere 2010:** Permite a los equipos de desarrollo colaborar fácilmente entre las plataformas. Team Explorer Everywhere 2010 contiene las herramientas y los plugins necesarios para acceder a Visual Studio Team Foundation Server 2010 desde dentro de los entornos basados en Eclipse, de manera que todo el mundo puede trabajar juntos y lograr los objetivos del negocio.

Dado el nivel de integración de funciones empresariales demandado a la herramienta, se empleó la edición de Visual Studio 2010 Premium para el desarrollo de este proyecto.

### **4.3 Syncfusion Essential Studio 2015**

El Essential Studio de Syncfusion es una suite de componentes y librerías para complementar el desarrollo en .Net que facilita el proceso de codificación



mediante componentes preconstruidos sobre patrones de arquitectura probados y las mejores prácticas de la industria (Syncfusion, Inc., 2016). Dentro de esta suite, se encuentra la biblioteca “XlsIO Essential”, que es una librería de interfaz de programación de aplicaciones (API por sus siglas en inglés) que permite leer y escribir archivos de Microsoft Excel al vuelo, empleando código .Net puro sin requerir interoperación con COM ni tener instalado Microsoft Office (Microsoft Office Automation), que fue empleada para el proceso de generación de la salida a Excel.

Cuenta con un modelo de objetos similar a las bibliotecas de Microsoft Office Automation. Puede ser utilizado en sistemas que no tienen instalado Microsoft Excel, por lo que es un excelente motor de archivos Excel para el generador de informes de datos tabulares. XlsIO esencial permite a los usuarios crear informes basados en documentos Excel que está disponible para todas las versiones de plataforma de .Net: Windows, WPF, ASP.NET WebForms, ASP.NET MVC, Silverlight, WinRT, Windows Phone, aplicaciones universales y Xamarin.



Ilustración 4, Logo de Syncfusion

Entre las principales características de la API de XlsIO Essential de Syncfusion nos encontramos con las siguientes:

- **Migración fácil de Ofimática:** Si se tiene código que ya está escrito utilizando la automatización de office, es muy fácil de migrar ese código para XlsIO desde las API son muy similares. Sin embargo, la similitud termina allí. XlsIO es más de 100 veces más rápido que la automatización de Office.

- **Compatible con versiones de Microsoft Excel 97 a 2013:** XlsIO esencial puede leer y escribir archivos en función de todas las variantes de formatos de archivo populares incluyendo Excel XLS y XLSX. Se puede incluso abrir un archivo en un formato y guardarlo de nuevo en otro formato.
- **Potente y completa:** Todos los elementos en un archivo de Excel típico como los valores de celda, formato, imágenes, gráficos y tablas dinámicas son accesibles a través de conjunto integral de funciones de la API XlsIO. Esto hace posible la creación de archivos con formato rico de Excel, así como extraer cualquier pieza deseada de la información en un archivo de Excel existente.
- **Fácil uso de los datos de importación y exportación:** XlsIO Essential ofrece varios métodos de utilidad para trabajar con fuentes de datos comunes. Sólo se necesita una línea de código para importar una matriz, tabla de datos, Data View, Columna de Datos y Business Objects en una hoja de cálculo.
- **Gráficas:** XlsIO Essential provee soporte avanzado para la creación y modificación de gráficos de Excel dentro de un libro. Los gráficos se pueden crear como un gráfico incrustado (gráfico está incrustado dentro de una hoja de trabajo) o como una hoja de cálculo gráfico (gráfico es una hoja de cálculo independiente). También provee varias opciones de formato para personalizar la apariencia de tales gráficos.
- **Tabla dinámica:** Cuando existe la necesidad de resumir informes grandes y complejos, Excel le permite resumir datos a través de tablas dinámicas. Un amplio conjunto de funciones del API están orientadas para crear y personalizar la apariencia y el comportamiento de las tablas dinámicas en Excel.
- **Convertir a CSV, PDF y HTML:** También es posible exportar el archivo de Excel a otros formatos comunes como CSV, PDF y HTML. La función de conversión a PDF y HTML está disponible para aplicaciones Windows, WPF, ASP.NET y ASP.NET MVC.

Microsoft Excel es la aplicación más utilizada de hoja de cálculo en el mundo, lo que hace de XLS / XLSX el formato de presentación de informes ideal para aplicaciones .NET. Algunas de las ventajas del uso de informes de Excel sobre otras alternativas como HTML y PDF son:

- El informe generado puede contener formatos ricos como Gráficas, imágenes, varias hojas de cálculo, fórmulas, tablas e incluso tablas dinámicas.
- El usuario final puede usar el poder de visualización de datos de MS Excel para manipular los informes generados. Por ejemplo, el usuario final puede generar varios gráficos para analizar y comprender las cifras de ventas en un informe de ventas de productos.

#### **4.4 Microsoft Entity Framework**

El Microsoft Entity Framework es la tecnología de acceso a datos de Microsoft .Net. Consiste un relacionador objeto-relacional (ORM por sus siglas en inglés: Object Relational Mapping) de .NET que facilita el desarrollo de aplicaciones de software orientadas a objetos, cuya información persistente se almacena en bases de datos relacionales (MSDN Entity Framework, 2015). Los arquitectos y programadores de aplicaciones orientadas a objetos se han enfrentado a la necesidad de lograr dos objetivos muy diferentes: Deben modelar las entidades, las relaciones y la lógica de los problemas empresariales que resuelven con entornos de programación orientados a objetos, y también deben luego trasladar la información gestionada de forma que pueda trabajar con los motores de datos que se usan para almacenar y recuperar la información, que en su mayoría son relacionales (RDBMS por sus siglas en inglés: Relational Data Base Management System).

Los datos pueden abarcar varios sistemas de almacenamiento, cada uno con sus propios protocolos; incluso las aplicaciones que funcionan con un único sistema de almacenamiento deben equilibrar los requisitos del sistema de almacenamiento con respecto a los requisitos de escribir un código de aplicación eficaz y fácil de mantener.

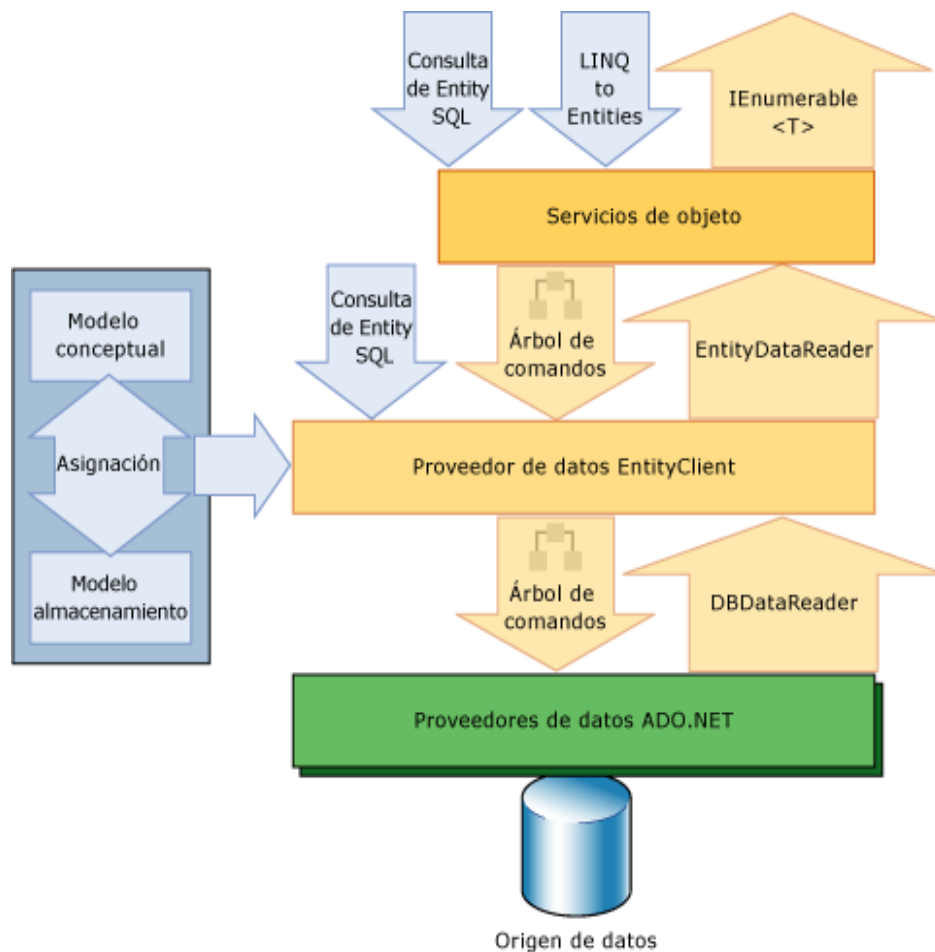


Ilustración 5: Componentes del MS Entity Framework

El Entity Framework permite a los desarrolladores trabajar con datos en forma de objetos y propiedades específicos del dominio, como clientes y direcciones de cliente, sin tener que preocuparse por las tablas y columnas de la base de datos subyacente donde se almacenan estos datos. Con Entity Framework, los desarrolladores pueden trabajar en un nivel mayor de abstracción cuando tratan con datos, y pueden crear y mantener aplicaciones orientadas a datos

con menos código que en las aplicaciones tradicionales. Dado que Entity Framework es un componente de .NET Framework, las aplicaciones de Entity Framework se pueden ejecutar en cualquier equipo en el que esté instalado .NET Framework a partir de la versión 3.5 SP1.

## 4.5 Bootstrap



Bootstrap, es un framework originalmente creado por Twitter (en el año 2011), que permite crear interfaces web con CSS y JavaScript que incluyen tipografías, botones, cuadros, menús y otros elementos que pueden ser utilizados en cualquier sitio web, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “**responsive design**” o diseño adaptativo.

Aunque el desarrollo del framework Bootstrap fue iniciado por Twitter, fue liberado bajo licencia MIT en el año 2011 y su desarrollo continua en un repositorio de GitHub.

Bootstrap es una excelente herramienta para crear interfaces de usuario limpias y totalmente adaptables a todo tipo de dispositivos y pantallas, sea cual sea su tamaño. Además, Bootstrap ofrece las herramientas necesarias para crear cualquier tipo de sitio web utilizando los estilos y elementos de sus librerías.

Desde la aparición de Bootstrap 3 el framework se ha vuelto bastante más compatible con desarrollo web responsive, entre otras características se han reforzado las siguientes:

- Soporte bastante bueno (casi completo) con HTML5 y CSS3, permitiendo ser usado de forma muy flexible para desarrollo web con unos excelentes resultados.
- Se ha añadido un sistema GRID que permite diseñar usando un GRID de 12 columnas donde se debe plasmar el contenido, con esto podemos desarrollar responsive de forma mucho más fácil e intuitiva.
- Bootstrap 3 establece Media Queries para 4 tamaños de dispositivos diferentes variando dependiendo del tamaño de su pantalla, estas Media Queries permiten desarrollar para dispositivos móviles y tablets de forma mucho más fácil.
- Bootstrap 3 también permite insertar imágenes responsive, es decir, con solo insertar la imagen con la clase “img-responsive” las imágenes se adaptaran al tamaño.

Todas estas características hacen que Bootstrap sea una excelente opción para desarrollar webs y aplicaciones web totalmente adaptables a cualquier tipo de dispositivo.

Bootstrap es compatible con la mayoría de navegadores web del mercado, y más desde la versión 3, actualmente es totalmente compatible con los siguientes navegadores:

- Google Chrome (en todas las plataformas).
- Safari (tanto en iOS como en Mac).
- Mozilla Firefox (en Mac y en Windows).
- Internet Explorer (en Windows y Windows Phone).
- Opera (en Windows y Mac).

### **Plug-ins de JavaScript**

Los componentes de JavaScript para Bootstrap están basados en la librería jQuery de JavaScript. Los plug-ins se encuentran en la herramienta de plug-in de jQuery. Proveen elementos adicionales de interfaz de usuario como diálogos,

tooltips y carruseles. También extienden la funcionalidad de algunos elementos de interfaz existentes, incluyendo por ejemplo una función de auto-completar para campos de entrada (input). La versión 3.0 soporta los siguientes plug-ins de JavaScript: Modal, Dropdown, Scrollspy, Tab, Tooltip, Popover, Alert, Button, Collapse, Carousel.

## 4.6 JQuery



Uno de los lenguajes de programación más importantes cuando hablamos de desarrollo web es sin duda alguna Javascript. Javascript es un lenguaje que se ejecuta a nivel cliente y el cual es

utilizado por la gran mayoría de las páginas web para desplegar contenido dinámico, hacer modificaciones automáticamente desde el navegador web de la persona y también ayuda a ejecutar tareas muy importantes tales como la validación de formularios.

jQuery es una librería JavaScript open-source, que funciona en múltiples navegadores, y que es compatible con CSS3. Su objetivo principal es hacer la programación “scripting” mucho más fácil y rápida del lado del cliente. Con jQuery se pueden producir páginas dinámicas así como animaciones parecidas a Flash en relativamente corto tiempo.

### **Origenes**

Creada inicialmente por John Resig, permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. El jQuery es la biblioteca de JavaScript más utilizada.

Doce meses después, en septiembre del 2007, jQuery ya mostraba una nueva interfaz de usuario y ya adquiría gran popularidad, y exactamente un año después, en septiembre 2008, Microsoft y Nokia anunciaron su soporte. Microsoft ha tratado de adoptar jQuery para usarlo en Visual Studio (integrándolo en el Framework AJAX de ASP.NET), y Nokia lo ha integrado en la plataforma de desarrollo de widgets.

Características.

- Selección de elementos DOM.
- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3
- Eventos.
- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.
- Animaciones personalizadas.
- AJAX.
- Soporta extensiones.
- Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.
- Compatible con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 10.6+ y Google Chrome 8+.

#### **4.6.1 Metodología de Desarrollo de Software**

El desarrollo del generador de reportes para el Banco de Proyectos del SNIP se realizó siguiendo la metodología estándar de desarrollo de software RUP (Rational Unified Process). Se seleccionó esta metodología por su exitosa trayectoria, definición formal, ser muy poderosa y poderse adaptar al contexto y las



necesidades de cualquier proyecto y organización. Esta capacidad de adaptación la hace idónea para su uso esta propuesta.

El “RUP (Proceso Unificado Racional - Proceso Unificado de Desarrollo de Software): Es un proceso que de manera ordenada defina las tareas y quién de los miembros del equipo de desarrollo las hará.”<sup>9</sup>

RUP es un proceso iterativo incremental que organiza el desarrollo del proyecto a través de cuatro fases: **inicio o conceptualización, elaboración, construcción y transición**, dentro de cada una de las cuales se pueden realizar una o más iteraciones, cada una con duración de dos a seis semanas. En cada iteración se atienden diferentes actividades correspondientes a las disciplinas de: a) Modelado del negocio, b) Relevamiento de los requisitos, c) Análisis y Diseño, d) Implementación, e) Pruebas y f) Despliegue; según se muestra en la siguiente ilustración:

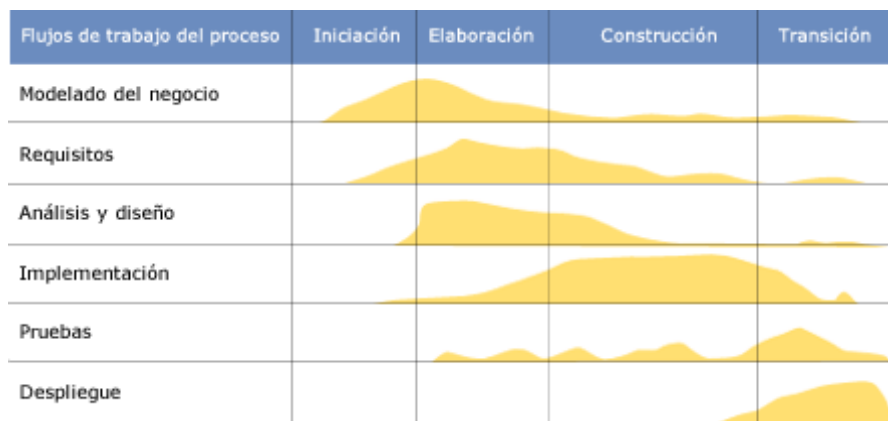


Ilustración 6: Flujo de trabajo del proceso

La documentación del proceso en RUP se realiza mediante diagramas estandarizados en el lenguaje de modelado UML.

<sup>9</sup> Ramos, I. A. *Herramienta Multimedia de apoyo a la Enseñanza de la Metodología RUP de Ingeniería del Software*. (pág. 7). Ver (Ramos, 2009)

En la fase de Conceptualización (Concepción o Inicio) se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.

En la fase de elaboración se define la arquitectura del sistema y se obtiene una arquitectura ejecutable que responde a los casos de uso críticos. Las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.

En la fase de construcción se realiza la codificación y pruebas de la máquina de generación y el utilitario de diseño de reportes.

Finalmente, en la fase de transición se implantan los productos y se capacitan a los usuarios para su uso correcto y productivo, asimismo, se identifica y corrige cualquier falla que haya pasado inadvertida en el proceso de construcción.

## **5 Desarrollo del Software Generador de Reportes**

El desarrollo del software que conforma la propuesta de solución, según se ha señalado en el diseño metodológico, se realizó siguiendo la metodología RUP, ajustada al alcance y propósito del presente trabajo. Esta metodología se basa en el proceso unificado, un proceso de desarrollo de software iterativo incremental que comprende cuatro fases: Inicio o Conceptualización, Elaboración, Construcción y Transición, Implantación o despliegue. En cada fase se pueden realizar de una a más iteraciones, que corresponde a unidades de ejecución con hitos o productos bien definidos. Cada iteración se programa con una duración de tiempo de dos a seis semanas.

### **5.1 Fase de Conceptualización**

Esta fase consiste en la definición de la visión general del proyecto, su alcance, el conocimiento del negocio, y la obtención de los requerimientos generales del proyecto.

#### **5.1.1 Requerimientos Generales**

El nuevo generador de reportes requerido para el Banco de Proyectos debe permitir a los usuarios generar reportes tabulares con salida a hojas de cálculo de Excel. Los reportes deben estructurarse en secciones lógicas. Para el alcance de este trabajo, se requieren al menos dos secciones: Encabezado y Cuerpo.

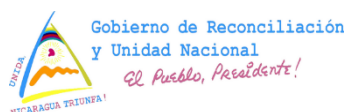
El encabezado debe tener en la esquina superior izquierda el logo del gobierno y en la centro, desde la primera fila del mismo, un listado de líneas de encabezado configurables por reporte, que predeterminadamente deben ser:

1. Denominación de la institución: **Ministerio de Hacienda y Crédito Público**
2. Denominación de la instancia: **Dirección General de Inversión Pública**

3. Nombre del Reporte, por ejemplo: [Programación Anual por Tipo de Recurso](#)
4. Especificación de la moneda de las cifras, por ejemplo: [Cifras en Córdoba](#).

La moneda puede ser Córdoba o Dólares.

El cuerpo del reporte se debe componer por una tabla (reporte tabular) estructurada por filas y columnas. El esquema de columnas es fijo y se debe definir cuando se diseña el reporte. Las columnas a la izquierda, normalmente las tres primeras, corresponden a la definición de los niveles de agregación de la información o variables del reporte, las cuales tienen un código, una descripción y una sigla opcionalmente. Las columnas a la derecha contienen los datos del reporte. Las filas corresponden a los registros específicos de la información del reporte con su descripción y datos. Estos se sumarían por cada nivel de agregación o variable que se seleccione. La tabla a continuación muestra un ejemplo de estructura de los reportes requeridos, con tres variables o niveles: Institución, Proyecto y Obra. Las columnas de datos, corresponden a los recursos financieros de cada registro.



#### MINISTERIO DE HACIENDA Y CREDITO PUBLICO

Dirección General de Inversión Pública

Programación Anual por Tipo de Recurso ( 2016 )

Cifras en Córdoba

Código	Descripción	Siglas	Recursos Internos		Recursos Externos		Total 2016
			Propios	Tesoro	Donación	Préstamo	
	<b>TOTAL</b>		<b>10,000,000</b>	<b>3,300,000</b>	<b>2,000,000</b>	<b>64,000,000</b>	<b>79,300,000</b>
	<b>INSTITUCIÓN 1</b>	<b>MTI</b>		<b>1,300,000</b>	<b>2,000,000</b>	<b>53,000,000</b>	<b>56,300,000</b>
	Proyecto 1	MTI		300,000		3,000,000	3,300,000
	Obra 1	MTI		100,000		1,000,000	1,100,000
	Obra 2	MTI		200,000		2,000,000	2,200,000
	Proyecto 2	MTI		1,000,000	2,000,000	50,000,000	53,000,000
	Obra 1	MTI		1,000,000	2,000,000	50,000,000	53,000,000
	<b>INSTITUCIÓN 2</b>	<b>ENACAL</b>	<b>10,000,000</b>	<b>2,000,000</b>		<b>11,000,000</b>	<b>23,000,000</b>
	Proyecto 1	ENACAL	10,000,000	2,000,000		11,000,000	23,000,000
	Obra 1	ENACAL	10,000,000	2,000,000		11,000,000	23,000,000

Opcionalmente, si se puede disponer de una sección de pie de página con información general como el número de página, por ejemplo.

Los reportes deben ser paramétricos, es decir, la naturaleza y organización de la información solicitada se puede especificar en función de parámetros que el usuario selecciona. Los parámetros para estos reportes corresponden a:

1. **Tipo de reporte** (naturaleza de los datos seleccionados, por ejemplo Financiero o de Indicadores Físicos),
2. **El esquema de columnas** (Cómo se estructura la información en las columnas de datos. Un reporte tabular consiste en una tabla con columnas fijas y agrupamiento de datos en las filas a los que se denominan niveles),
3. **La selección de grupos o niveles de agrupación.** (variables por las que se pueden realizar agregados de los datos).

Asimismo se pueden seleccionar parámetros de filtrado de datos como son:

4. **La versión de los datos** (El Banco de Proyectos permite manejar múltiples versiones de los mismos datos para la realización de simulaciones y proyecciones), y
5. **El período de tiempo**, que puede ser un Año, un período de años consecutivos, o un período de meses consecutivos.

A través de esta serie de parámetros, un mismo reporte puede generar múltiples salidas con diferentes formas de ordenación y vistas de los mismos datos. Por ejemplo, un reporte financiero de la programación anual puede estar agrupado por [<Institución, Proyecto y Fuente de Financiamiento>](#) con lo cual se proporciona una vista de distribución institucional de la inversión identificando a detalle los proyectos específicos y cómo se financian. El mismo reporte agrupado por [<Sector Económico, Fuente de Financiamiento>](#) genera una vista totalmente diferente de la inversión por destino de gasto: Salud, Educación, Infraestructura Vial, etcétera, y cómo se financia la inversión en cada sector por la fuente (Si son recursos del gobierno, si son donaciones o préstamos y los organismos que los proveen).

Para la implementación práctica del Generador de reportes se requiere que la solución comprenda tres componentes:

1. **Herramienta Diseñadora de Reportes:** Aplicación que permite al usuario elaborar su reporte y publicarlo.
2. **La Máquina Generadora de Reportes:** Conjunto de clases de .Net. encargadas de recibir una solicitud de generación de reporte, con sus parámetros, y deberá generarlo y entregarlo al servidor de reportes.
3. **El Servidor de Reportes:** Aplicación con interfaz gráfica con la que interactuará el usuario para elegir el reporte, seleccionar los parámetros y solicitar la generación del mismo, la cual se lo entregará en formato Excel.

### **5.1.2 Reglas del negocio**

Las siguientes reglas abarcan algunas de las condiciones que se deben de cumplir para el correcto funcionamiento de los que es la maquina generadora de reportes y la herramienta diseñadora de reporte, estas reglas son fundamentales para el entendimiento de las funcionalidades de la aplicaciones.

1. Las funciones base se tienen que nombrar de la siguiente manera: con el prefijo “FTRep” seguido de la descripción que la acción que realiza.
2. Para la creación de niveles se tiene que nombrar el código con el prefijo “Cod” y la descripción con el prefijo “Des”, seguido del nombre del nivel que se desea asignar.
3. La máscara de los niveles siempre es de 6 dígitos ej.: “000000”

### **5.1.3 Requerimientos Específicos por Aplicativo**

#### ***5.1.3.1 Requerimientos Funcionales***

El sistema debe permitir lo siguiente:

- **Herramienta Diseñadora de Reportes:**
  - Introducir datos o la modificación de los mismos para la creación de reporte que necesite el usuario final.
  - Clonar reporte como base para un nuevo reporte similar al anterior.

- Editar reporte existente.
- Introducir datos o la modificación de los mismos para la creación de parámetros.
- Modificar parámetros.
- **Máquina de Generación de Reportes:**
  - Recibir la solicitud de generación del reporte y los parámetros seleccionados.
  - Generar el código SQL necesario a partir del diseño del reporte y los parámetros seleccionados por el usuario y ejecutarlo contra la base de datos.
  - Entregar los datos del reporte listos para formateo al servidor de reportes.
- **Servidor de Reportes:**
  - Presentar al usuario los diferentes tipos de reportes existentes en una interfaz web sencilla para que éste pueda seleccionar uno de ellos.
  - Desplegar los reportes del tipo seleccionado por el usuario para que éste elija el reporte que desea ejecutar.
  - Permitir al usuario seleccionar o editar los parámetros del reporte, según sea el tipo de parámetro.
  - Solicitar a la Máquina de Generación de reportes la generación del reporte.
  - Formatear el reporte en el formato de trabajo de la institución (.xlsx) y entregarlo al usuario.

#### **5.1.3.2 Requerimientos No Funcionales**

Entre las principales consideraciones de requerimientos no funcionales hechas por el cliente del sistema (Dirección de Desarrollo del Banco de Proyectos de la DGIP) destacan:

- Que el Diseñador de reportes sea una aplicación de escritorio Windows. Que el Servidor de Reportes sea una aplicación Web ASP.Net (Web Forms) y su Máquina de Generación de Reportes sea una clase .Net.
- Que el Diseñador de reportes sea desarrollado para operar en sistema operativo Windows 7 o superior. Que el Servidor de Reportes y su Máquina de Generación de Reportes sean desarrollados para operar sobre el sistema operativo Windows 2008 R2, sobre el servidor de aplicaciones de Microsoft Internet Information Server (IIS) con Framework .Net V 3.5 o superior.
- El despliegue en producción debe alojar las bases de datos requeridas en Microsoft SQL Server 2005.
- Utilizar Microsoft Visual Studio 2010 para la implementación del proyecto.
- Debe de estar orientado a trabajar sobre una arquitectura de red centralizada y de dominio público sobre protocolo HTTP y ser accedida para cualquier usuario del SNIP.
- El servidor de reportes no debe implementar seguridad porque se requiere que todos los reportes sean públicos.

#### **5.1.4 Estudio de Factibilidad**

A fines de determinar si la propuesta de solución era pertinente y posible de realizar, amén de determinar el costo de la misma, se realizó un estudio de factibilidad enfocando diferentes aspectos del proceso, tales como la factibilidad técnica que contempla las necesidades del usuario y las necesidades para el desarrollo del software; la factibilidad operativa, que enfoca las capacidades requeridas para operar las diferentes herramientas que se ocupan para el acceso de las aplicaciones; la factibilidad económica, que contempla los costos de desarrollo y el costo de los equipos; y finalmente la factibilidad legal que contempla las leyes que implican el desarrollo de la aplicación y las limitaciones legales que afectaría el alcance del generador de reportes y la forma de hacer las cosas.



#### **5.1.4.1 Factibilidad técnica**

En este estudio se verifica la factibilidad para la implementación del generador de reportes. Se establecen los requisitos mínimos en cuanto a recursos de hardware y software que serían necesarios para su desarrollo y correcto funcionamiento, y se analizan los posibles riesgos que se pueden presentar en el tiempo de ejecución del sistema, así como las medidas preventivas para evitar que ocurran estos riesgos y las medidas correctivas en caso de presentarse.

#### **➤ Requisitos Mínimos del Sistema**

Para el correcto funcionamiento del Generador de Reportes y para aprovechar las capacidades e información contenida dentro de su registro de base de datos se sugiere:

##### **1. Equipo de Hardware:**

- **Para la Herramienta Diseñadora de Reportes:** En vista que es una aplicación desktop de carácter cliente – servidor que se opera desde una PC o portátil, se requieren las siguientes características mínimas:
  - Procesador de tipo Intel core i3 a 1.2 GHz o superior y 2 GB de memoria RAM. Los archivos locales ocupan menos de 50 MB en disco local, puesto que la información se almacena en el servidor de base de datos.
- **Para el Servidor de Reportes:** El servidor de reportes consiste en la aplicación que permite al usuario seleccionar un reporte, editar los parámetros y generar el reporte. Requiere servicio de infraestructura empresarial web y dos roles de servidores, físicos o virtuales con las siguientes características mínimas:
  - **Servidor de aplicaciones:** Servidor web equipado con IIS y tecnología del Framework .Net 3.5 o superior. Servidor de clase

empresarial con Procesador de al menos 2.5 GHz, Memoria principal operativa de 8 GB en RAM y almacenamiento de al menos 80 GB.

- **Servidor de Base de datos:** Servidor de clase empresarial con Procesador de al menos 2.5 GHz, Memoria principal operativa de 8 GB en RAM y almacenamiento de al menos 120 GB.
- **Para ejecutar los reportes: Cliente de Reportes:** Los usuarios finales emplearán la aplicación del servidor de reportes a través del Web, para lo cual requieren acceso al Internet y un equipo con las condiciones mínimas para navegar en internet: Procesador contemporáneo de al menos 1 GHz de frecuencia, 1 GB de RAM, 20 MB de disco duro libre para temporales de navegación, y conexión WiFi o Ethernet con conectividad activa a Internet.

## **2. Software del desarrollador**

Debido a restricciones impuestas por el cliente en los requerimientos no funcionales, se utilizarán:

- SQL Server 2008 R2 (compatible con base de datos 2005 que será requerido para el despliegue)
- Visual Studio 2010 professional edition.
- Sistema Operativo Windows 7 o superior.
- Syncfusion Essential Studio 2015, para la generación a Excel.
- Navegador de Internet: Internet Explorer 7 o superior, Google Chrome o Mozilla Firefox, para pruebas de funcionamiento.
- Microsoft Office, OpenOffice, LibreOffice, Kingsoft Office o cualquier otra aplicación para gestionar archivos de Excel, para pruebas de funcionamiento.

## **4. Software del Usuario**

Para la ejecución del sistema se requieren los siguientes elementos de software instalados y operando en la computadora del usuario:

- Sistema operativo Windows 7 o superior.
- Navegador de Internet: Internet Explorer 7 o superior, Google Chrome o Mozilla Firefox.
- Microsoft Office, OpenOffice, LibreOffice, Kingsoft Office o cualquier otra aplicación para gestionar archivos de Excel.

### ➤ **Riesgos del Sistema, Prevención y Mitigación**

Como todo sistema de información basado en computadora, existen múltiples factores de riesgo a la operación técnica del sistema en producción. A continuación se listan los más relevantes, incluyendo las medidas de prevención y acciones de mitigación en caso de darse los eventos de riesgo que comprometan la disponibilidad del sistema:

Riesgo	Medidas de Prevención	Acciones de Mitigación
Que se presenten fallas en la operación del sistema.	1. Realizar pruebas exhaustivas durante el proceso de construcción para asegurar la calidad del software y código lo más posible libre de defectos.	1. Establecer mecanismos expeditos para atender las incidencias de parte de los usuarios y contar con personal para realizar mantenimiento al software según sea requerido.
Que no esté disponible el sistema para su operación. Esto puede ser principalmente por: 1. Falla en la conectividad al Internet 2. Falta de energía eléctrica. 3. Servidor bajo por falla de condiciones técnicas y ambientales. 4. Servidor bajo por ataque informático. 5. Servidor bajo por falla en sistema operativo.	1. Seleccionar medios y proveedores de servicio de Internet confiables. 2. Contar con UPS online y Planta de respaldo de energía con relevamiento automático ante fallas. 3. Contar con certificación técnica y climatización adecuada en el centro de datos. 4. Asegurar los mecanismos y medidas de seguridad de red.	1. Recurrir al servicio de emergencia del ISP. 2. Hacer llamada de emergencia a proveedores de mantenimiento de UPS y Planta. 3. Hacer llamada de emergencia a proveedores de mantenimiento de climatización. 4. Activar procedimientos de recuperación y forense.

Riesgo	Medidas de Prevención	Acciones de Mitigación
6. Aplicación baja por falla en servidor de aplicaciones u otras dependencias.	5. Mantener al día las actualizaciones del SO.	5. Reiniciar SO, o activar servidor contingente.
7. Base de datos fuera de línea	6. Mantener monitoreo de funcionamiento.	6. Reiniciar aplicaciones, SO o activar servidor contingente.
	7. Mantener sistema de respaldo y recuperación de base de datos.	7. Reiniciar servicio, reiniciar servidor, activar proceso de recuperación.

➤ **Verificación de Factibilidad Técnica para el Generador de reportes en DGIP**

Se realizó proceso de verificación del cumplimiento de los requisitos de hardware, software y ante riesgos de operación del sistema, obteniéndose los siguientes resultados:

**1. Equipo de Hardware:**

- Los equipos de cómputo con que cuenta la DGIP actualmente superan las especificaciones mínimas de hardware.
- El equipo que empleará el usuario de DGIP que operará la **Herramienta Diseñadora de Reportes** consiste en una computadora de escritorio optimizada para desarrollo con procesador Intel core i7 de 4 núcleos a 3.6 GHz, 8 GB de memoria RAM y 500 GB en disco duro. Asimismo se dispone de portátiles con procesador Intel core i7 de 4 núcleos a 2.4 GHz, 4 GB de memoria RAM y 500 GB en disco duro.
- El **Servidor de aplicaciones** con que se dispone es un Servidor web equipado con IIS 7.0 y tecnología del Framework .Net 4.0. Es un servidor de clase empresarial con Procesador Intel Xeon de 4 núcleos a 2.5GHz x 2 con HyperThreading activado para un total de 8 núcleos y 16 procesos simultáneos. Memoria principal operativa de 16 GB en RAM y almacenamiento redundante basado en 4 discos SAS de 15Krpm en RAID 5 con espacio de 516 GB para datos.

- El **Servidor de Base de datos** es un servidor de clase empresarial con Procesador Intel Xeon de 4 núcleos a 2.5GHz x 2 con HiperThreading activado para un total de 8 núcleos y 16 procesos simultáneos. Memoria principal operativa de 16 GB en RAM y almacenamiento redundante basado en 5 discos SAS de 15Krpm en RAID 5 con espacio de 688 GB para datos.

## **2. Software del desarrollador**

DGIP cuenta con suficientes licencias de desarrollo de todos los softwares requeridos y proporcionará en calidad de préstamo dichas licencias a los autores durante el período de desarrollo del generador de reportes.

## **3. Software del Usuario**

Asimismo, los equipos de los usuarios cuentan con todo el software requerido para ejecutar el sistema de manera óptima.

## **4. Preparación ante Riesgos**

- **Fallas de Software:** Se realizará la codificación del sistema empleando las técnicas de calidad de ingeniería de software así como las recomendaciones de mejores prácticas de la industria para asegurar en lo posible código de calidad y libre de defectos. Ante el evento de incidencias de fallas por errores de software, DGIP tiene instaurado un mecanismo y procedimiento expeditos para atender las incidencias y un equipo de desarrollo de planta para corregirlas en muy corto tiempo.
- **Falla en la conectividad:** DGIP cuenta con un servicio corporativo de Internet con suficiente capacidad de servicio para operar su propio nodo (el nodo snip.gob.ni es alojado *inhouse* en las instalaciones de DGIP). Tiene un acuerdo de servicio de primer nivel con atención prioritaria por parte del centro de operaciones de red (NOC) del proveedor de servicios de Internet (ISP).

- **Falta de energía eléctrica:** Se dispone de medidas de respaldo eléctrico para apoyar la disponibilidad de los servicios y continuidad del negocio ante falla del fluido eléctrico. Un primer nivel consiste en una UPS central de respaldo del datacenter del tipo online con salida senoidal pura y regulada. El segundo nivel consiste en sistema de planta o generador de respaldo con encendido y transferencia automática en caso de falla de la corriente eléctrica comercial. Se asegura el combustible suficiente para este generador que tiene autonomía de 12 horas continuas sin relleno de combustible, pero puede funcionar de régimen si fuera necesario. Tanto la UPS como el generador cuentan con contratos de mantenimiento rutinario y de atención de emergencias.
- **Servidor bajo por falla de condiciones técnicas y ambientales:** El centro de datos de DGIP cuenta con las condiciones eléctricas adecuadas, su sistema eléctrico está certificado, debidamente polarizado y con un esquema de centros de carga jerárquico. Se cuenta con condiciones de temperatura entre 15° C y 25° C, para garantizar un mejor aprovechamiento y seguridad del equipo tanto de servidores como de usuarios. El centro de datos de DGIP se mantiene a 18° C como promedio, para garantizar el estado óptimo de los servidores. El área de trabajo de usuarios se mantiene a 22° C en promedio. Se tiene un contrato de mantenimiento del sistema de climatización con atención ante emergencias.
- **Servidor bajo por ataque informático:** A fin de reducir la posibilidad de una intrusión que vulnere la operación de los sistemas o la confidencialidad de los datos, DGIP integra un sistema de seguridad informática basada en servidores de gestión unificada de seguridad ante amenazas (UTM por sus siglas en inglés: Unified Threat Management) que integra todos los mecanismos de seguridad de redes la industria contemporánea. Este equipamiento se complementa con un conjunto de políticas de uso de servicios y de seguridad que proporcionan un nivel de protección apropiado de acuerdo para los estándares actuales.

- **Servidor bajo por falla en sistema operativo:** Por política informática, la DDBP de DGIP, mantiene un proceso de actualización periódica del sistema operativo de los servidores. Asimismo, se llevan y monitorean de forma regular los registros de eventos y transacciones (LOGs) para dar seguimiento al funcionamiento de los servidores.
- **Aplicación baja por falla en servidor de aplicaciones u otras dependencias:** DGIP tiene asimismo, políticas para mantener actualizadas las aplicaciones de servicios, de acuerdo a las recomendaciones de los diferentes fabricantes. Asimismo, se monitorean los registros de eventos y de transacciones de los mismos.
- **Base de datos fuera de línea:** Se mantiene activado un sistema de respaldo automático de datos tanto de la aplicación como de la Base de Datos. Se considera este sistema suficiente para los requerimientos de la aplicación. El sistema implementado es redundante y multimedio: El primer nivel de respaldo es a un servidor de respaldos (en disco duro y en línea). El segundo nivel de respaldo es a cartucho de disco extraíble (en diferido, según programación). El tercer nivel de respaldo es manual en DVD realizado mensualmente con copia local y externa (fuera del edificio).

El resultado del proceso de verificación, concluyó, por tanto que DGIP tiene cubiertos todos los requerimientos técnicos para hacer factible el generador de reportes.

#### ***5.1.4.2 Factibilidad Operativa***

Para favorecer el uso masivo del generador de reportes por parte de todos los usuarios del SNIP, será requerido desarrollar una aplicación lo más amigable posible, de tal forma que sin mucha dificultad el usuario pueda adaptarse y aprovechar al máximo las facilidades que ésta le brinde, ahorrando tiempo y permitiendo la realización de otras actividades.

El sistema funcionará en Internet, al cual se accederá a través de la página oficial de los sistemas del SNIP (<http://www.snip.gob.ni>). Los usuarios podrán visualizar la información que ellos soliciten a través de la sección de reportes en donde se presentará una interfaz agradable y sencilla. El usuario deberá de seleccionar diferentes opciones para la generación exitosa del reporte. Cualquier usuario con experiencia de navegar páginas dinámicas en Internet podrá operar el generador, por cuanto el diseño de la página se realizará siguiendo los estándares de Web 2, empleando la librería de interfaz de usuario más común hoy día: BootStrap<sup>10</sup> por lo cual la organización de las páginas y la distribución de los controles será muy familiar a usuarios de aplicaciones web y móviles.

Por su parte, la aplicación de escritorio para la creación y modificación de reportes estará desarrollada con los patrones de interfaz más comunes, por ejemplo el estilo que introdujo Outlook, que implementa barras de herramientas con botones gráficos amplios que capturan la atención de los usuarios. Adicionalmente, esta herramienta será usada por un profesional informático que es el principal encargado del manejo de reportes, por lo cual no se espera que exista ninguna limitante operativa en su uso.

DGIP y sus directivos se encuentran anuentes a aceptar los cambios y mejoras que el sistema ofrezca dentro del entorno de su organización, llevando a la conclusión de que el sistema es factible operativamente, ya que se cuenta con la aceptación y la tecnología para desarrollar el sistema con éxito.

#### **5.1.4.3 Factibilidad económica**

DGIP cuenta con el equipamiento, las licencias de software requeridos para implementar la propuesta de solución, así como la plataforma de servicios para su publicación y presupuesto para mantener los costos recurrentes de la misma, por un lado. El costo de desarrollo de software será asumido por los autores del

---

<sup>10</sup> Bootstrap es la más popular biblioteca existente hoy día para creación de aplicaciones HTML y Javascript adaptables a múltiples formatos. Ver referencia: (Bootstrap, 2016)



estudio, por el otro, con lo cual todos los costos económicos están cubiertos, haciendo factible el generador de reportes desde esta perspectiva.

Sin embargo, para fines ilustrativos para clientes que deseen implementar un generador de reportes similar y que no cuenten con la base económica de DGIP, se calculan los costos de implementar la solución, obviando los costos recurrentes debido a lo difícil de éste cálculo, al no contar con información de depreciación del equipo, ni estadísticas de consumo energético, ni de recursos de mantenimiento, etcétera. Cabe mencionar que éstos se podrían contabilizar si se asumen mediante una suscripción de alojamiento, las cuales suelen partir de USD 15 (quince dólares) mensuales.

### **Costo del desarrollo del Generador de Reportes**

✓ Insumos:

✓ Equipamiento Cómputo:

- 1 servidor de aplicaciones: USD 10, 000
- 1 servidor de base de datos: USD 12, 000
- 1 PC de desarrollo: USD 1, 500
- 1 PC de Diseño reportes: USD 1, 000

TOTAL: **USD 24, 500**

✓ Software de desarrollo:

- Visual Studio Professional : USD 500
- SQL Server Estándar (1 core): USD 5, 000
- Syncfusion Suite: USD 1, 000

TOTAL: **USD 6, 500**

✓ Software de desarrollo, Alternativa Community :

- Visual Studio Community : USD 0
- SQL Express\*: USD 0
- Sincfusion Suite Community: USD 0

TOTAL: **USD 0**

\*SQL Express es gratuito para desarrollo. Deberá pagarse licencia estándar, enterprise, o servicio de nube en SQL Azure para la publicación y producción.

✓ Construcción:

2 desarrolladores de sistemas por 3 meses a USD 1,300 mensual:

○ Desarrolladores:  $2 \times 3 \times 1300 =$  USD 7,800

TOTAL: **USD 7,800**

**Costos Totales del Generador del Reportes:**

- a) Para una organización que parte de cero (incluyendo equipamiento, software de desarrollo licenciado y desarrollo de software): **USD 38,800**
- b) Para una organización que ya tiene un sistema web desarrollado sobre .Net con base de datos SQL Server (incluye solo desarrollo): **USD 7,800**

**5.1.4.4 Factibilidad Legal**

En el presente trabajo de diploma para implementar el generador de reportes se respeta y se cumple con la ley de los derechos de autor, respetando todas las prerrogativas que dicha ley establece. DGIP en su función de órgano rector del SNIP, ya cuenta con licencias de uso para cada software a emplearse para el desarrollo de los aplicativos. Asimismo, cuenta con la base legal que le otorga el derecho y la obligación del desarrollo de los sistemas informáticos de Sistema Nacional de Inversión Pública. Dicha base legal se establece en la Ley No. 550, Ley de Administración Financiera y del Régimen Presupuestario de la República de Nicaragua, en los artículos 164; 166, literal h; y 175, que corresponde, éste último, al Banco de Proyectos:

*Artículo 175.- Banco de Proyectos. Es el registro oficial único de todas las iniciativas de inversión del Sector Público en las distintas etapas del ciclo de proyectos, incluyendo las ideas de proyectos, aquellos en etapa de estudio de pre*

*inversión, los que han iniciado gestión de ingresos, los que inician o iniciaron su ejecución, y los que han finalizado su ejecución y se encuentran en la fase de operación.*

*Los organismos y entidades que integran el Sistema Nacional de Inversiones Públicas deben incorporar al Banco de Proyectos todos sus proyectos con la correspondiente información requerida por la Unidad de Inversiones Pública. Las instituciones serán responsables de la calidad y oportunidad de la misma, considerándose ésta de interés nacional.*

## **5.2 Fase de Elaboración**

En la fase de elaboración se realiza el análisis y diseño de la aplicación, y se obtiene la base de arquitectura de la misma. Por el alcance en tamaño y complejidad de este proceso de desarrollo, que es moderado, en línea con lo planteado por Larman (Larman, 2003), se circunscribe el análisis en esta fase al desarrollo de la actividad de Implementación de Casos de Uso y Modelado Conceptual de la aplicación. Por su cuenta, para el diseño se consideran las actividades de Diseño de Clases, y Modelado de Interacción, mediante Diagrama de Secuencia. Asimismo, por ser un producto de software orientado a datos (reportes) se realizó el modelado del diseño de la Base de Datos.

### **5.2.1 Modelado de Casos de Uso**

El modelo de Casos de Uso representa la forma en cómo un Cliente (Actor) usa la aplicación, además de la forma, tipo y orden en como los elementos interactúan.

Un diagrama de casos de uso consta de los siguientes elementos:

- Actor
- Casos de Uso.
- Relaciones de Uso, Herencia y Comunicación.

## 5.2.2 Casos de Uso para la Herramienta Diseñadora de Reportes

### 5.2.2.1 Diagrama de Casos de Uso

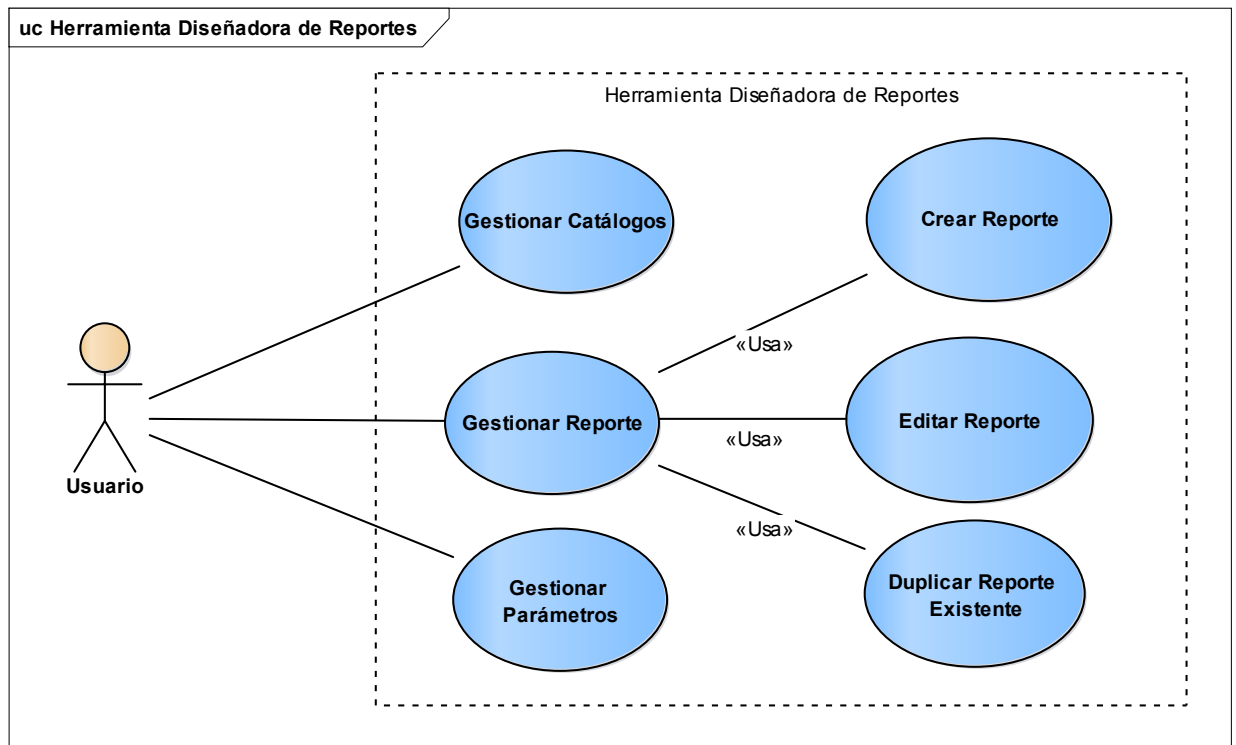


Ilustración 7: Casos de uso de la Herramienta Generadora de Reportes

### 5.2.2.2 Especificación de Casos de Uso

Nombre:	<b>Gestionar Catálogos</b>	
Autor:	Paul Zamora	
Fecha:	19/12/2015	
Descripción:	Se actualizan o se crean nuevos niveles o categorías de reportes según sea la selección del usuario.	
Actores:	Usuario	
Precondiciones:	No hay.	
<b>Flujo Normal</b>		
<b>Usuario</b>		<b>Herramienta</b>
Selecciona la opción “Gestionar Catálogo” para mantener los elementos de dicho catálogo (creación o modificación).		Presenta las opciones de catálogos existentes (Nivel y Categoría).

Elige, según sea el caso (Nivel o Categoría), el catálogo al cual desea dar mantenimiento.	Lista los elementos existentes en el catálogo seleccionado por el usuario.
Si lo que desea es agregar un nuevo elemento al catálogo, elige la opción "Agregar".	Presenta formulario con campos vacíos para que el usuario provea la información del nuevo elemento.
Llena los campos con la información requerida y selecciona "Guardar".	Guarda la información del nuevo elemento del catálogo en la base de datos.
Si lo que desea es modificar un elemento existente, busca y selecciona el elemento a editar.	Presenta formulario con campos llenos con la información actual del elemento para que el usuario pueda editarla.
Edita los campos específicos con la información a actualizar y selecciona "Guardar".	Actualiza la información del elemento en la base de datos.
<b>Flujo Alternativo</b>	
<b>Usuario</b>	<b>Herramienta</b>
Usuario selecciona salir sin guardar.	No guarda el nuevo elemento o no actualiza los valores modificados por el usuario en un elemento existente.

Nombre:	<b>Gestionar Reporte</b>
Autor:	Paul Zamora
Fecha:	19/12/2015
Descripción:	Permite crear, editar o duplicar un reporte
Actores:	Usuario
Precondiciones:	Las propias de los Casos de Usos invocados.
<b>Flujo Normal</b>	
<b>Usuario</b>	<b>Herramienta</b>
Selecciona opción "Crear Reporte"	Invoca al caso de Uso "Crear Reporte"
Selecciona opción "Duplicar Reporte"	Invoca al caso de Uso "Duplicar Reporte"
Selecciona opción "Editar Reporte"	Invoca al caso de Uso "Crear Reporte"
<b>Flujo Alternativo</b>	
<b>Usuario</b>	<b>Herramienta</b>
Usuario selecciona salir del programa.	El programa termina y no muestra ninguna información.

Nombre:	<b>Crear Reporte</b>	
Autor:	Paul Zamora	
Fecha:	19/12/2015	
Descripción:	Permite la creación de un nuevo reporte.	
Actores:	Usuario	
Precondiciones:	La función SQL base del reporte como fuente de datos.	
<b>Flujo Normal</b>		
<b>Usuario</b>		<b>Herramienta</b>
<Ha seleccionado “Crear Reporte” en caso de uso que invoca>		Presenta lista de funciones SQL disponibles como base para reporte.
Elige la función SQL base que requiere emplear.		Carga niveles de agrupación y las columnas de datos que contiene la función SQL base.  Solicita completar la información propia del reporte.
Provee la información requerida: nombre del reporte, descripción, categoría del reporte, niveles predeterminados y columnas con su orden que tendrá el reporte.  Elige guardar el reporte.		Guarda la información del reporte en la base de datos.
<b>Flujo Alternativo</b>		
<b>Usuario</b>		<b>Herramienta</b>
Usuario selecciona salir sin guardar.		No guarda el reporte nuevo y termina.

Nombre:	Duplicar Reporte
Autor:	Paul Zamora
Fecha:	19/12/2015
Descripción:	Permite crear un nuevo esquema de reporte a partir de una copia exacta de uno existente (duplicado o clonado) para luego modificarlo, facilitando así no tener que crear un reporte desde cero si existe uno similar.
Actores:	Usuario
Precondiciones:	Esquema de reportes modelo existente.
Flujo Normal	
Usuario	Herramienta
<Ha seleccionado “Duplicar Reporte” en caso de uso que invoca>	Presenta listado de reportes.

Elige el reporte que requiere duplicar.	Carga campos de información general de identificación del Reporte para su definición por parte del usuario.
Define el nombre del reporte, Descripción, y si lo desea cambiar niveles predeterminados y columnas con su orden.  Elige guardar el reporte.	Guarda la información del reporte en la base de datos.
<b>Flujo Alternativo</b>	
<b>Usuario</b>	<b>Herramienta</b>
Usuario selecciona salir sin guardar.	No guarda el reporte nuevo duplicado.

Nombre:	<b>Editar Reporte</b>
Autor:	Paul Zamora
Fecha:	19/12/2015
Descripción:	Permite la edición o actualización de un esquema de reporte.
Actores:	Usuario
Precondiciones:	Esquema de reporte a modificar existente.
<b>Flujo Normal</b>	
<b>Usuario</b>	<b>Herramienta</b>
<Ha seleccionado “Editar Reporte” en caso de uso que invoca>	Lista los esquemas de reportes existentes.
Selecciona esquema de reporte.	Carga formulario con campos llenos con la información del esquema de reporte seleccionado.
Edita o modifica la información del esquema de reporte que sea requerida: nombre al reporte, descripción, categoría del reporte, niveles predeterminados y columnas con su orden.  Elige guardar el reporte.	Guarda la información del reporte en la base de datos.

<b>Flujo Alternativo</b>	
<b>Usuario</b>	<b>Herramienta</b>
Usuario selecciona salir sin guardar.	No guarda los cambios al reporte seleccionado.

Nombre:	<b>Gestionar Parámetros</b>
Autor:	Paul Zamora
Fecha:	19/12/2015
Descripción:	Permite identificar los parámetros requeridos por las funciones SQL base de reportes y mapearlos a controles de interfaz gráfica para sistematizar su invocación en la base de datos.
Actores:	Usuario
Precondiciones:	Función SQL existente en la base de datos.

<b>Flujo Normal</b>	
<b>Usuario</b>	<b>Herramienta</b>
Solicita "Configuración de Parámetros".	Presenta las funciones SQL disponibles.
Si desea crear la configuración de parámetros para una nueva función selecciona la opción "Nueva Función".	Presenta las funciones SQL de la base de datos que no están registradas.
Selecciona la Función SQL a la cual se va a crear la configuración de parámetros.	Presenta los parámetros de la función con las opciones de mapeo.
Selecciona por cada parámetro el tipo de dato, tipo de control, nombre de control.  Elige guardar parámetros.	Guarda la información de la nueva Función con su Configuración de Parámetros en la base de datos.
Si lo que desea es modificar los parámetros de una función existente, selecciona la Función en el listado y elige "Modificar Parámetros".	Presenta los parámetros de la función con las opciones de mapeo.
Selecciona por cada parámetro el tipo de dato, tipo de control, nombre de control.  Elige guardar parámetros.	Guarda la información de la nueva Función con su Configuración de Parámetros en la base de datos.

<b>Flujo Alternativo</b>	
<b>Usuario</b>	<b>Herramienta</b>



Usuario selecciona salir sin guardar.	No guarda la información de la nueva Función con su Configuración de Parámetro o no actualiza los datos de la Configuración de Parámetros, según sea lo solicitado.
---------------------------------------	---

### 5.2.3 Casos de Uso del Servidor de Reportes

#### 5.2.3.1 Diagrama de Casos de Uso

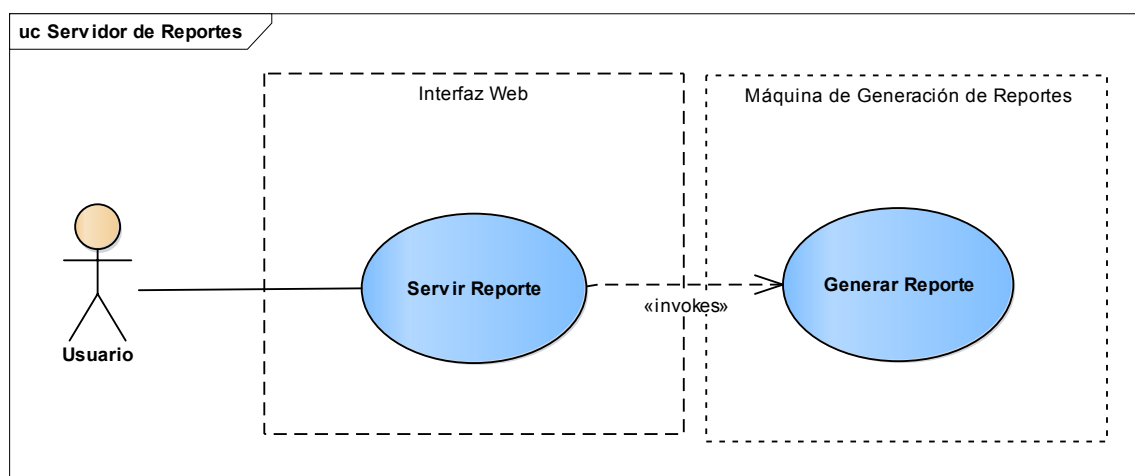


Ilustración 8: Casos de Uso del Servidor de Reportes

#### 5.2.3.2 Especificación de Casos de Uso

<b>Nombre:</b>	<b>Servir Reporte</b>
Autor:	Paul Zamora
Fecha:	19/12/2015
Descripción:	Permite al usuario elegir un reporte, seleccionar sus parámetros y generarlo a través del Internet.
Actores:	Usuario
Precondiciones:	Acceso al Internet. Reporte requerido existente publicado en el Servidor.
<b>Flujo Normal</b>	
<b>Usuario</b>	<b>Herramienta</b>
Solicita la opción "Generar reporte".	Presenta listado de Categorías de Reportes existentes.

Selecciona una Categoría de Reporte.	Lista los diferentes Reportes existentes para la categoría seleccionada.
Selecciona el Reporte que desea generar.	Carga campos con los parámetros del reporte con valores predeterminados para que el usuario ratifique o modifique dichos valores: Niveles, Filtros y Parámetros de Tiempo.
Selecciona los parámetros del reporte deseados: jerarquía de niveles de agrupación que compondrán el reporte, fuentes de datos, período de tiempo para el reporte. Selecciona la opción “Generar”	<p>Invoca al caso de uso “Generar Reporte” de la Máquina Generadora de reportes.</p> <p>Recibe de éste, los datos del reporte.</p> <p>Formatea los datos del reporte en archivo de Excel y lo envía al usuario.</p>
<b>Flujo Alternativo</b>	
<b>Usuario</b>	<b>Herramienta</b>
Usuario selecciona “Cancelar” o “Salir”	No genera reporte alguno.

<b>Nombre:</b>	<b>Generar Reporte</b>
Autor:	Paul Zamora
Fecha:	19/12/2015
Descripción:	Genera script para obtener los datos del reporte organizado, agrupado y agregado en los diferentes niveles que solicitó el usuario, lo ejecuta contra la base de datos y regresa el set de datos resultado listo para su formateo y entrega.
Actores:	Caso de Uso Servir Reporte que lo invoca
Precondiciones:	Ninguna
<b>Flujo Normal</b>	
<b>Servir Reporte</b>	<b>Generar Reporte</b>
Invoca generar reporte, enviando identificación del reporte y parámetros seleccionados.	<p>Genera el script para obtener los datos del reporte de la función base, organizados, agrupados y agregados en los diferentes niveles que establecen los parámetros, y filtrado por los parámetros de filtro.</p> <p>Ejecuta el script SQL contra la base de datos y recibe el set de datos resultado.</p>

	Entrega el set de datos resultado al invocador.
<b>Flujo Alternativo</b>	
<b>Servir Reporte</b>	<b>Generar Reporte</b>
	Si ocurre un error, no se retorna dato alguno. Se genera una excepción.

#### **5.2.4 Modelado conceptual o de Dominio**

El modelo de dominio es un modelo conceptual de alto nivel que define los objetos del mundo físico (clases conceptuales) que son significativos en el dominio del problema. Es una representación de las clases conceptuales del mundo real, no de los componentes de software. Documenta la relación entre las clases conceptuales y sus responsabilidades (Larman, 2003).

Utilizando la notación UML, un modelo del dominio se representa con un diagrama de clases en el cual a las mismas no se le definen ni atributos ni operaciones. Las clases en el diagrama representan las entidades físicas del dominio del problema que se relacionan entre sí. Estas relaciones se detallan para hacerlas más explícitas, mediante etiquetas y notación de cardinalidad.

##### **5.2.4.1 *Información base del Dominio del Problema***

El objeto del Generador de Reportes es el Reporte. Un reporte se clasifica en una única Categoría de Reporte, de acuerdo a la naturaleza de sus datos (Financiero o de Indicadores Físicos, por ejemplo). El alcance del reporte dentro del universo de datos almacenados está definido por sus parámetros de filtrado, los que pueden ser del tipo “Serie de Datos” (datos oficiales o datos de proyecciones o ejercicios de simulación), o de “Tiempo” (Año, Conjunto de años o Período de meses consecutivos). Existen otros filtros por “Variable o Dimensión” que

quedaron fuera del alcance de este trabajo de diploma, de acuerdo a lo establecido por el cliente (DDBP) en las sesiones de definición del alcance del Generador de Reportes.

El Reporte se estructura en dos Secciones lógicas: La sección de Encabezado del Reporte y la Sección de Datos<sup>11</sup>. La Sección de Encabezado se compone de una serie de líneas de Encabezado con información general (institución, nombre, etc.). Dado que la herramienta solicitada debe generar reportes del tipo tabular, la Sección de Datos se estructura como una tabla lógica organizada verticalmente en Niveles y horizontalmente en Columnas. El reporte toma la información base de una Función SQL que define las Columnas y los Niveles.

Los Niveles corresponden a las diferentes variables por las cuales se realizarán los agrupamientos de los datos, por tanto los niveles tienen un orden que el usuario debe poder establecer.

Cada reporte debe tener un conjunto de Niveles Predeterminados con su orden preestablecido de acuerdo a la estructura de agrupamiento más usada, a fin de facilitar la selección de este parámetro.

---

<sup>11</sup> Opcionalmente se puede agregar las secciones de Encabezado de Página y Pie de Página, pero no se incorporarán en este proyecto.

### 5.2.4.2 Diagrama del modelo de dominio

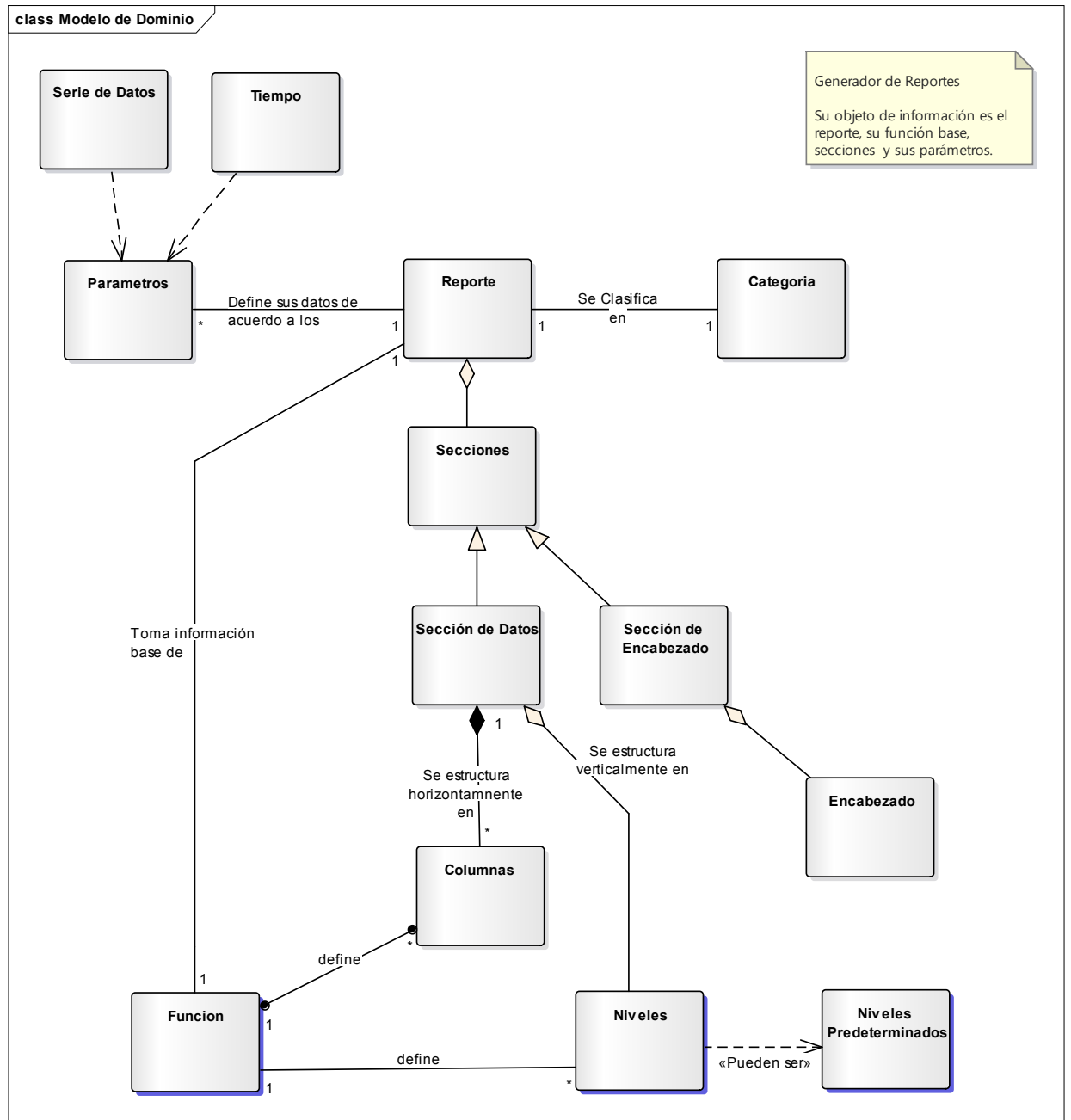


Ilustración 9: Modelo de Dominio del Generador de Reportes

### 5.2.5 Diagrama de Clases

El diagrama de clases es el diagrama principal para el diseño de la aplicación, éste presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para atributos (datos de las clases) y operaciones (métodos de las clases para comunicarse entre sí).

El diagrama de clases se deriva del modelo de dominio del problema, el cual se refina para determinar cuáles sustantivos de la información base del dominio del problema son clases del dominio y cuales son solo atributos o estructuración lógica del problema. Para el desarrollo del Generador de Reportes, a partir de su diagrama de dominio se pudieron identificar las siguientes clases:

1. **CategoriaReporte:** Representa las diferentes categorías en que pueden estar clasificados los reportes.
2. **Reporte:** Representa el reporte en sí.
3. **Encabezado:** Representa a las diferentes líneas de Encabezado que puede tener el reporte.
4. **OrigenDatos:** Representa a las Funciones SQL que proveen los datos al reporte, y las operaciones con la base de datos para invocarla.
5. **NivelReporte:** Representa a los diferentes Niveles que puede tener el reporte y que en la ejecución de la aplicación fueron seleccionados por el usuario para la generación del reporte específico.

#### **Relaciones entre las clases:**

- La clase Reporte está asociada a la clase Encabezado en que uno o muchos reportes tienen asignado un encabezado.
- La clase Reporte está asociada a la clase NivelReporte en que uno o muchos reportes pueden contener muchos niveles de agrupación.
- La clase OrigenDatos y la clase Reporte tienen una relación de uno a muchos en la que una Función puede ser utilizada como origen de datos en muchos reportes.

- clase CategoriaReporte y la clase Reporte tienen una relación de uno a muchos en el que una categoría puede estar en muchos reportes.

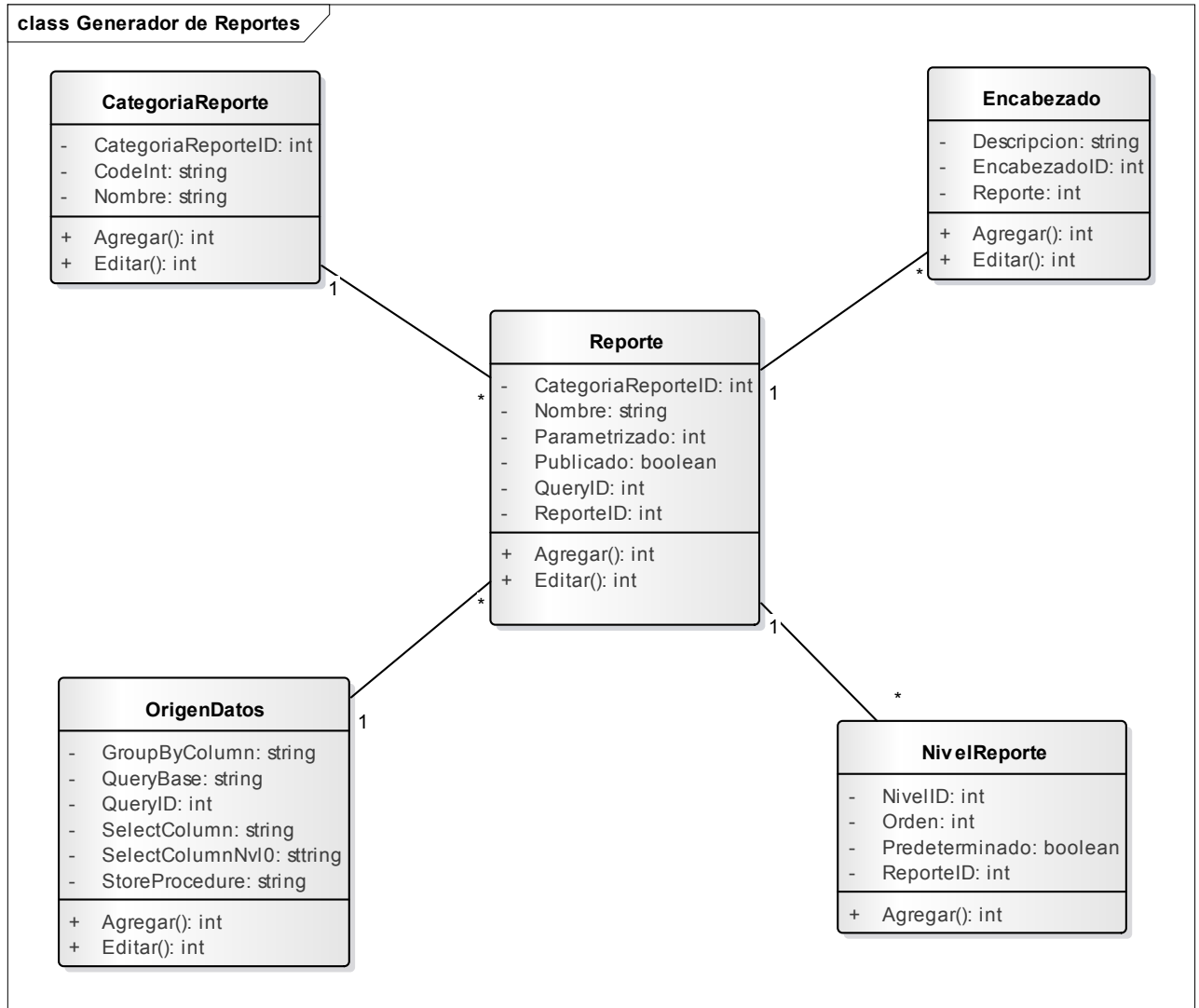


Ilustración 10: Diagrama de Clases del Generador de Reportes

## 5.2.6 Diseño de base de datos

A partir del diagrama de dominio y del diagrama de clases, se determinó la estructura de base de datos que sería requerida para almacenar la información de los reportes, a fin que la Máquina Generador pudiera producir, a partir de la misma,

conjugada con los parámetros seleccionados, el script SQL para obtener los datos del reporte de la Base de Datos del sistema.

Cabe mencionar que se decidió diseñar la aplicación y la base de datos del generador de reportes de forma tal que pudiera usarse integrada a la base de datos del sistema (preferentemente como un esquema) o de forma separada (como otra base de datos). Asimismo, se agregaron campos de control de interfaz uso actual o futuro (a solicitud de DGIP) para manejo de seguridad.

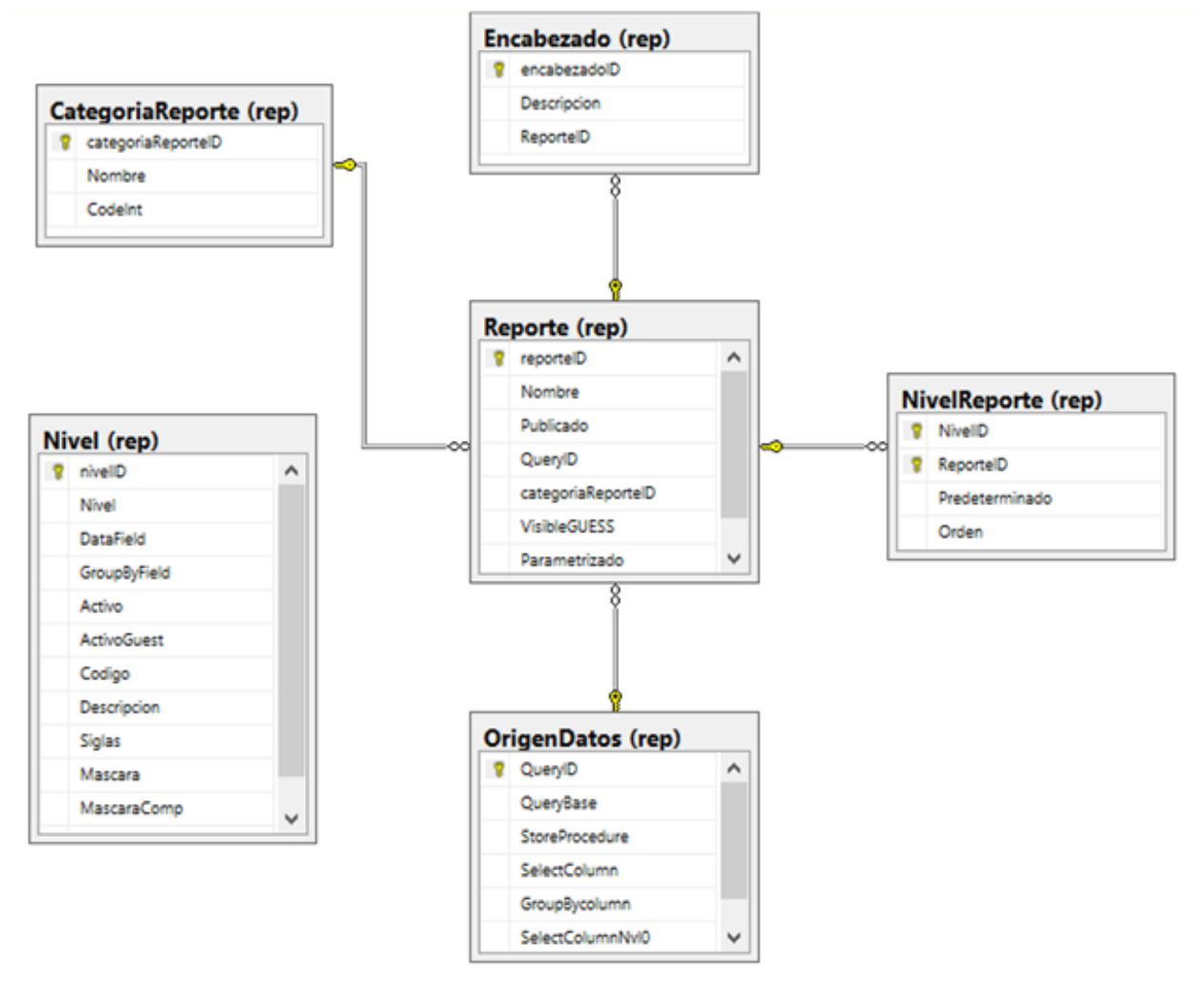


Ilustración 11: Diseño de la Base de Datos del Generador de Reportes.



## 5.2.7 Diagrama de Actividades

En UML un diagrama de actividades se usa para mostrar la secuencia de actividades realizadas en flujo de trabajo, detallando las rutas de decisiones que existen en el progreso de eventos.

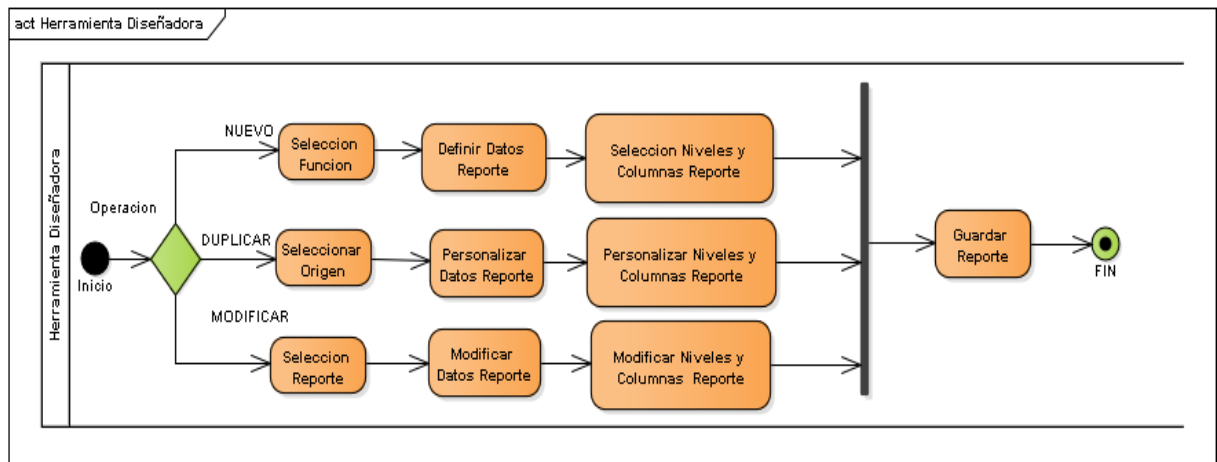


Ilustración 12: Diagrama de Actividades del Diseñador de Reporte

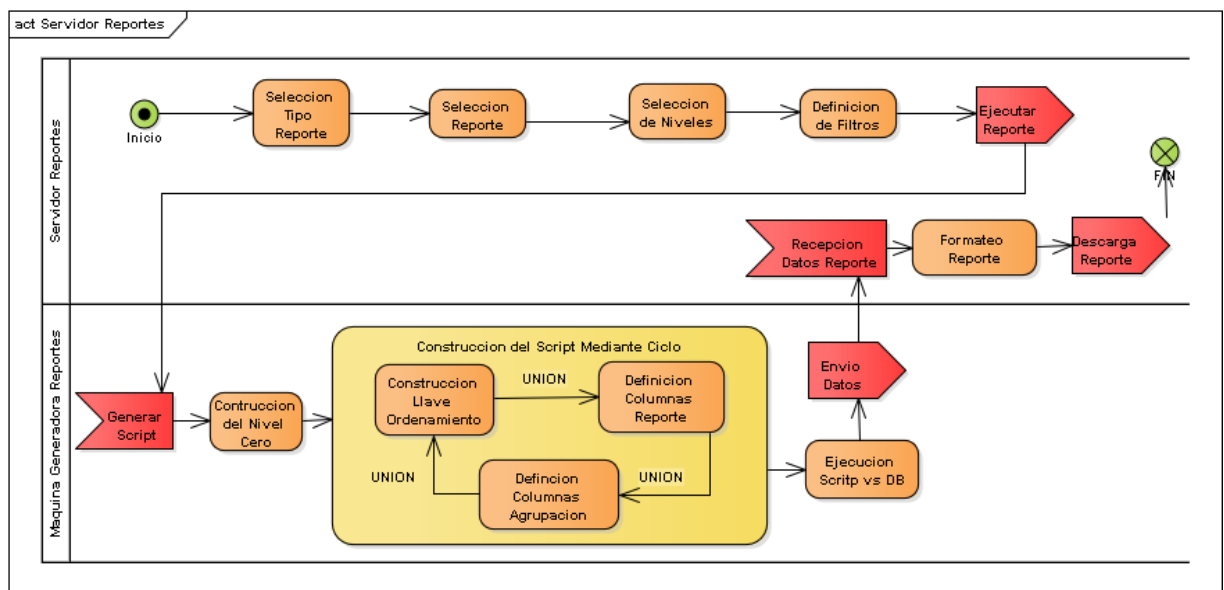


Ilustración 13: Diagrama de Actividades del Servidor de Reportes

## 5.2.8 Diagrama de Secuencia

El diagrama de secuencia muestra los objetos participantes en las interacción de la ejecución del aplicativo y los mensajes intercambiados (Larman, 2003).

A continuación se presentan los diagramas de secuencia del Generador de Reportes:

### 1. Herramienta Diseñadora de Reportes:

- Creación de un Nuevo Reporte.
- Edición de un Reporte.
- Duplicación de un Reporte.
- Configuración de los Parámetros de un Reporte.
- Creación de un nuevo ítem de Catálogo.
- Edición de un ítem de Catálogo.

### 2. Servidor de Reportes:

- Generación de Reporte.

#### 5.2.8.1 Diagramas de Secuencia de la Herramienta Diseñadora de Reportes

#### 5.2.8.2 Creación de Reporte

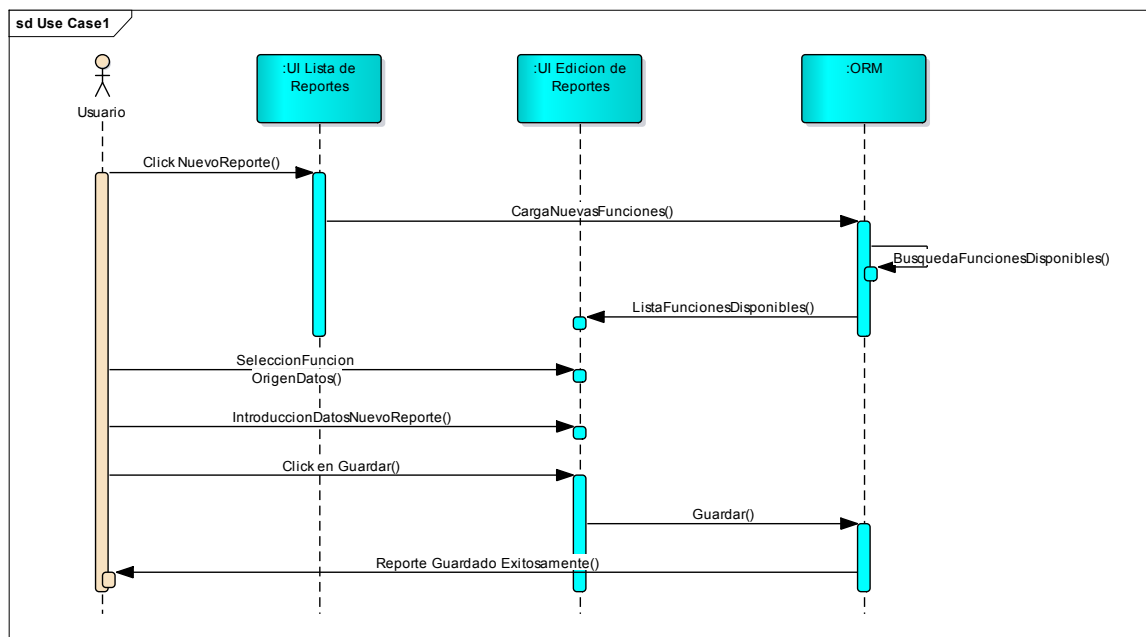


Ilustración 14: Diagrama de Secuencia Creación de Reporte

### **Secuencia de Acciones:**

El Usuario presiona el botón Nuevo Reporte en la interfaz UI Lista de Reportes, la que invoca a la interfaz UI Edición de Reportes, que realiza la invocación del método que obtiene las funciones disponibles para crear un reporte, que se presentan al usuario por parte de UI Lista de Reportes.

El usuario introduce los datos referentes al nuevo reporte que incluyen la selección de la función que desea utilizar como origen de datos, los niveles de agrupación que desea se carguen de forma predeterminada, las columnas que incluirá el nuevo reporte, la categoría a la que pertenece el reporte, el nombre del reporte y el encabezado del reporte.

El usuario da la orden de guardar el nuevo reporte, con lo que se envía los datos a la base de datos y luego se envía un mensaje de que el reporte se ha Guardado Exitosamente.

### **5.2.8.3 Edición de Reporte**

Ver diagrama de secuencia en página siguiente.

### **Secuencia de Acciones:**

El Usuario selecciona el reporte que desea editar y presiona el botón Modificar Reporte en la interfaz UI Lista de Reportes, automáticamente ésta invoca a la interfaz de UI Edición de Reportes, a continuación se realiza la invocación del método que obtiene todos los datos que corresponden al reporte seleccionado y se cargan los datos en la interfaz.

El usuario modifica los datos del reporte que desea actualizar tales como las columnas que este presenta, los niveles de agrupación predeterminados, la categoría a la que pertenece, el nombre del reporte y el encabezado.

El usuario activa el botón guardar para actualizar los datos, con lo que se envía los datos a la base de datos y luego se envía un mensaje de que el reporte se ha Actualizado Exitosamente.

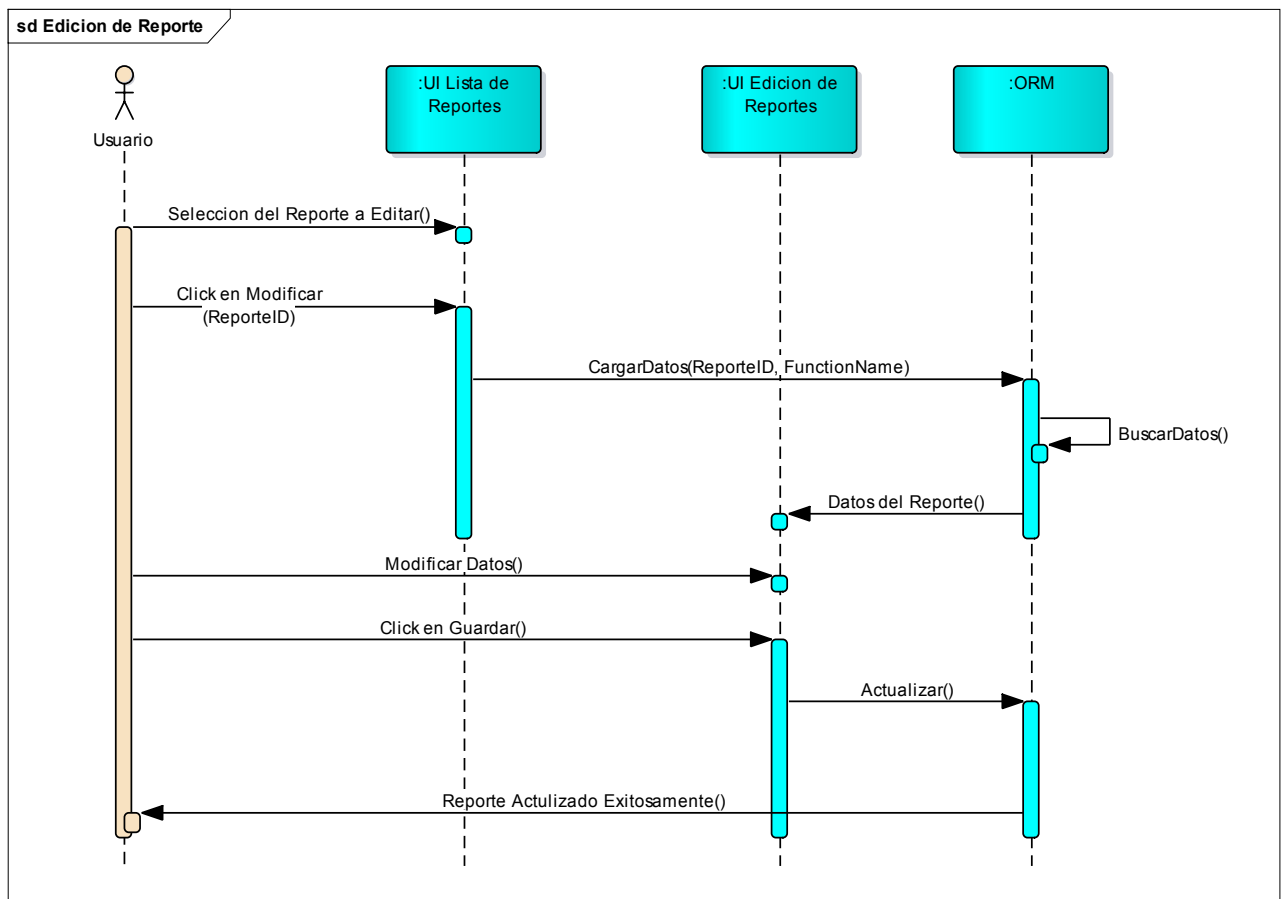


Ilustración 15: Diagrama de Secuencia de operación Edición de Reporte.

#### 5.2.8.4 Duplicar Reporte

Ver diagrama de secuencia en página siguiente.

#### Secuencia de Acciones:

El Usuario selecciona el reporte que desea duplicar y presiona el botón Duplicar en la interfaz UI Lista de Reportes, automáticamente ésta invoca a la interfaz de

UI Edición de Reportes, que a continuación se realiza la invocación del método que obtiene todos los datos que corresponden al reporte seleccionado.

El usuario introduce los datos referentes al nuevo reporte que incluyen la selección de las columnas que este presenta, de los niveles de agrupación que desea sean predeterminados, la categoría a la que pertenece, el nombre del reporte y el encabezado.

El usuario activa el botón guardar, con lo que se envía los datos a la base de datos y luego se envía un mensaje de que el reporte se ha Guardado Exitosamente.

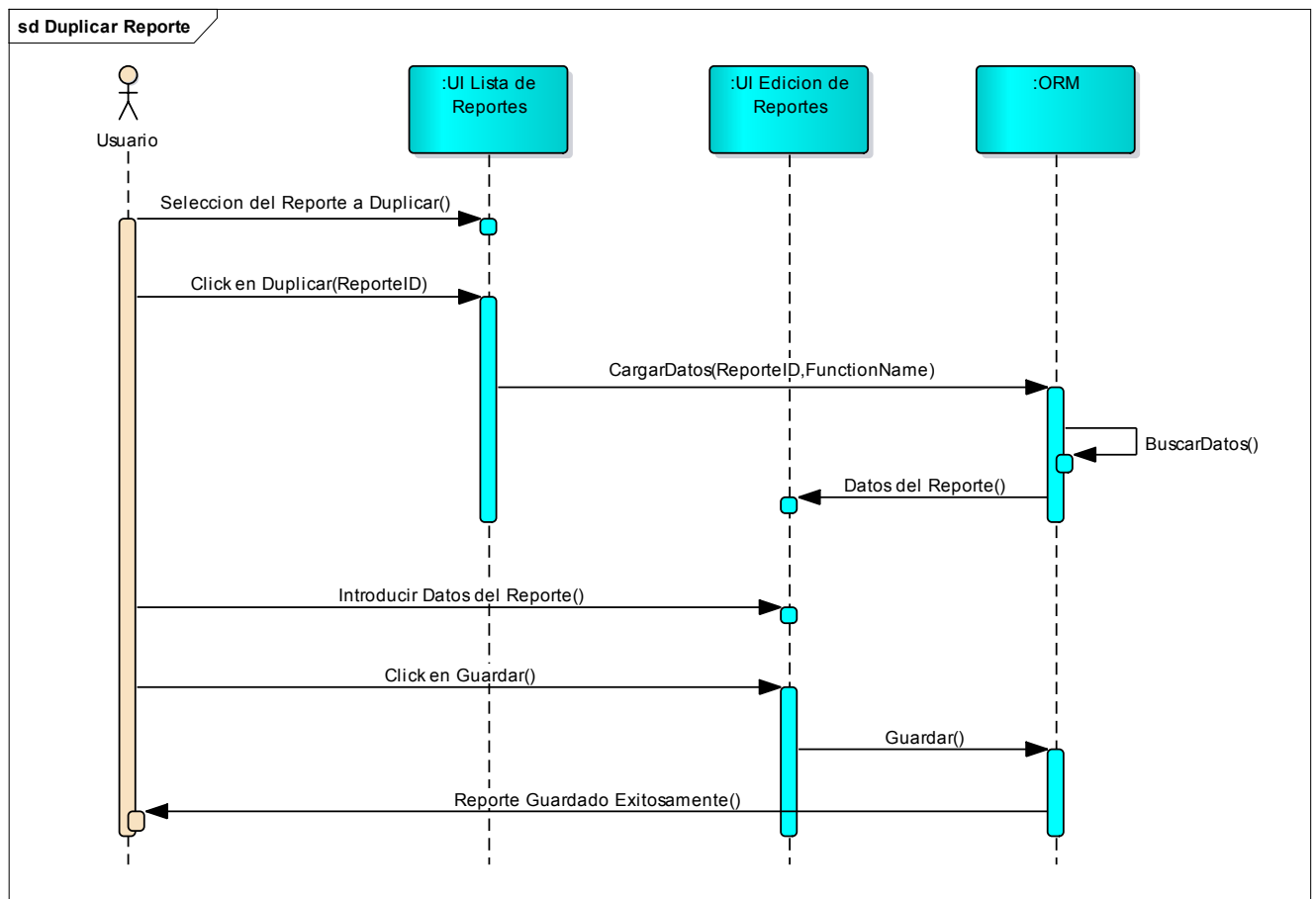


Ilustración 16: Diagrama de secuencia de la operación Duplicar Reporte.

#### 5.2.8.5 Configuración de Parámetro de Reporte

Ver diagrama de secuencia en página siguiente.

## Secuencia de Acciones:

El Usuario selecciona la función que desea parametrizar y presiona el botón en la interfaz UI Configuración de Parámetros, que automáticamente invoca a la interfaz de UI Configuración Detalle, a continuación se realiza la invocación del método que obtiene la metadata de la función que se desea parametrizar.

El usuario realiza el proceso de parametrización, que incluye la definición de la correlación entre cada parámetro de la función con un control en la interfaz web, definiéndole el nombre y tipo de control.

El usuario presiona el botón guardar, en este momento se envía los datos de la parametrización a la base de datos y luego se envía un mensaje de que el Parámetro se ha Guardado Exitosamente.

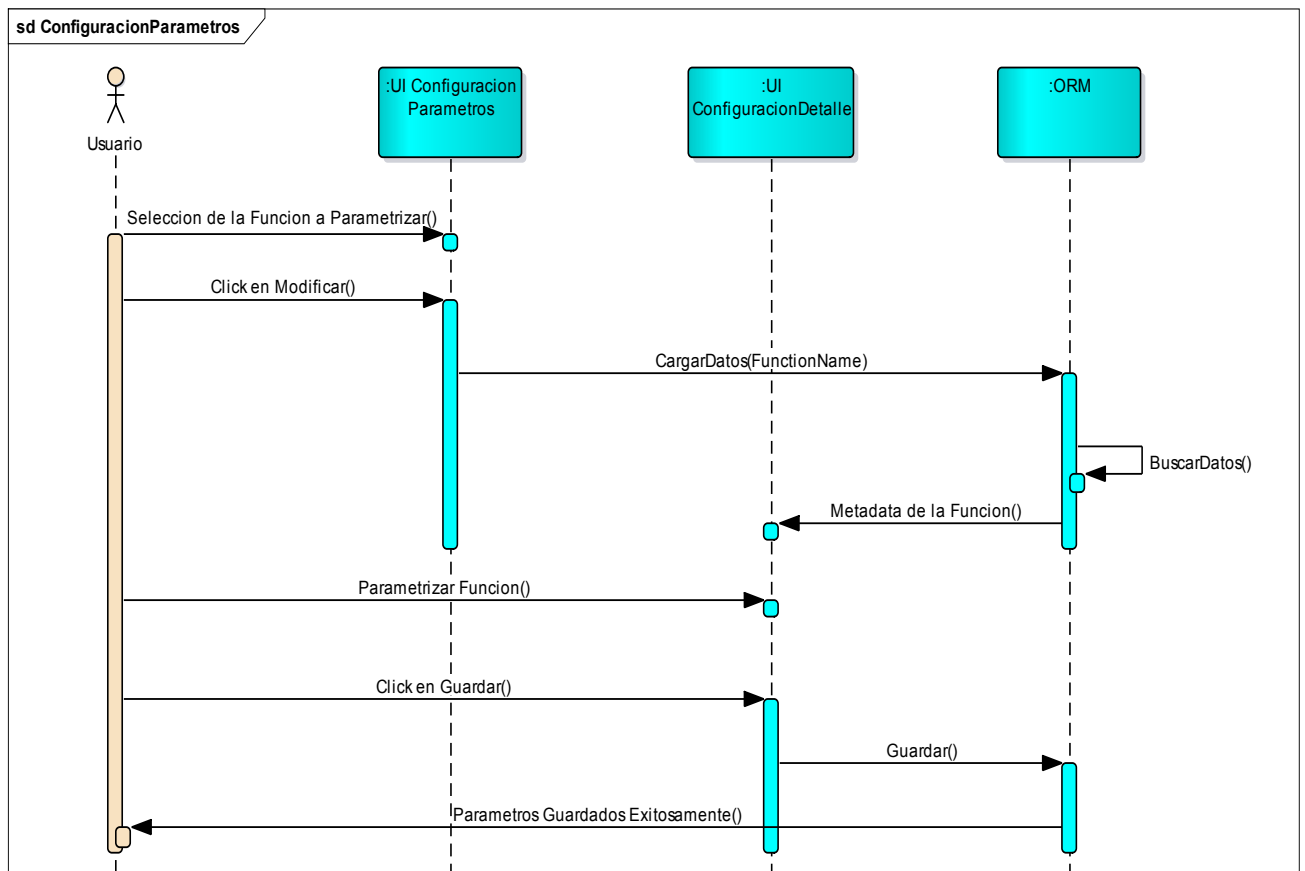


Ilustración 17: Diagrama de secuencia de la operación Configuración de Parámetros.

### 5.2.8.6 Creación de un nuevo ítem de Catálogo

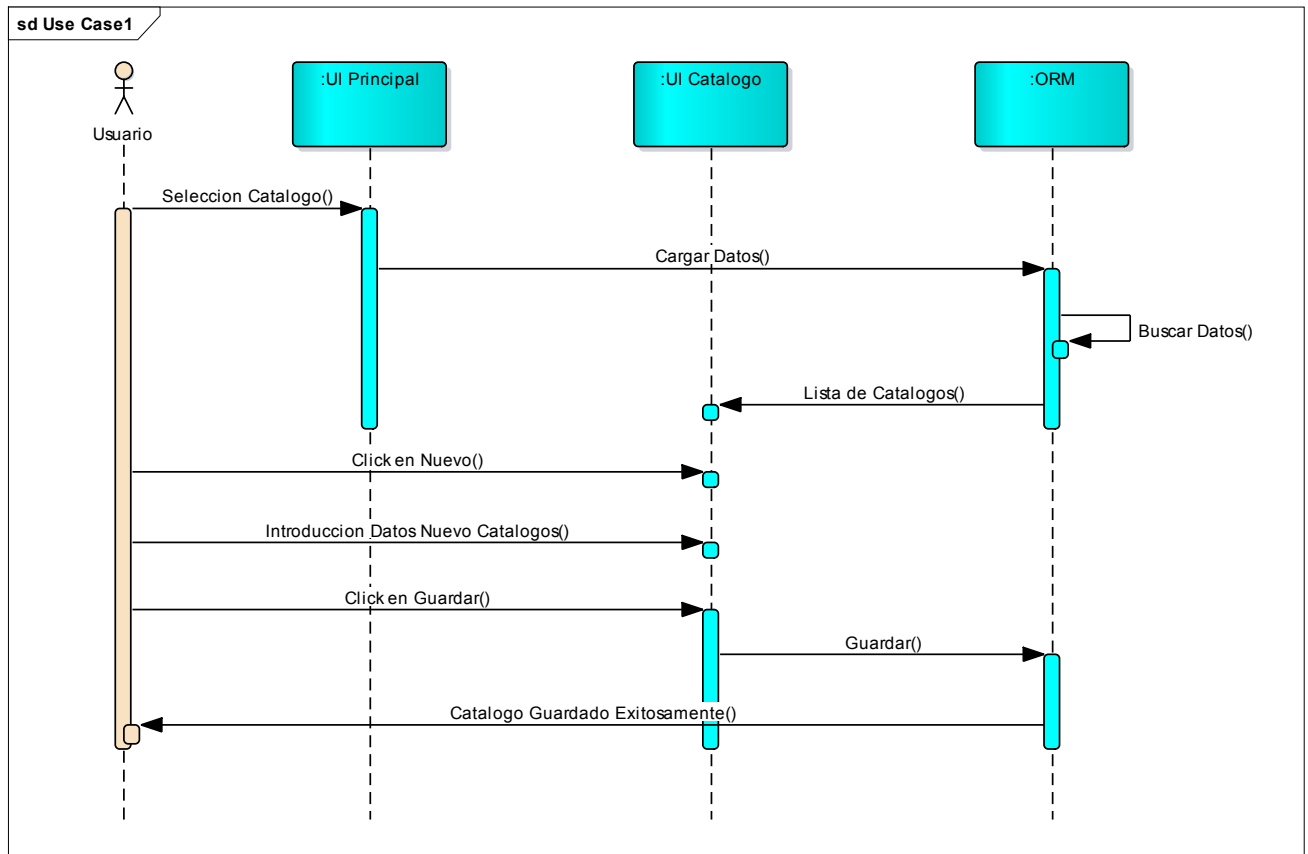


Ilustración 18, Diagrama de secuencia creación de catálogo de categorías o niveles

#### Secuencia de Acciones:

El usuario selecciona el tipo de Catálogo que desea mantener (Categoría o Niveles) desde la interfaz UI Principal, automáticamente manda a llamar la interfaz que corresponde al catálogo seleccionado, en seguida se realiza la invocación del método que carga la lista de catálogos.

El usuario presiona el botón nuevo desde la interfaz UI Catalogo, introduce los datos del nuevo ítem de catálogo y presiona el botón Guardar. Se envían los datos del nuevo catálogo a la base de datos, luego se envía un mensaje de que el Ítem de Catálogo se ha Guardado Exitosamente.

### 5.2.8.7 Edición de Ítem de Catálogo

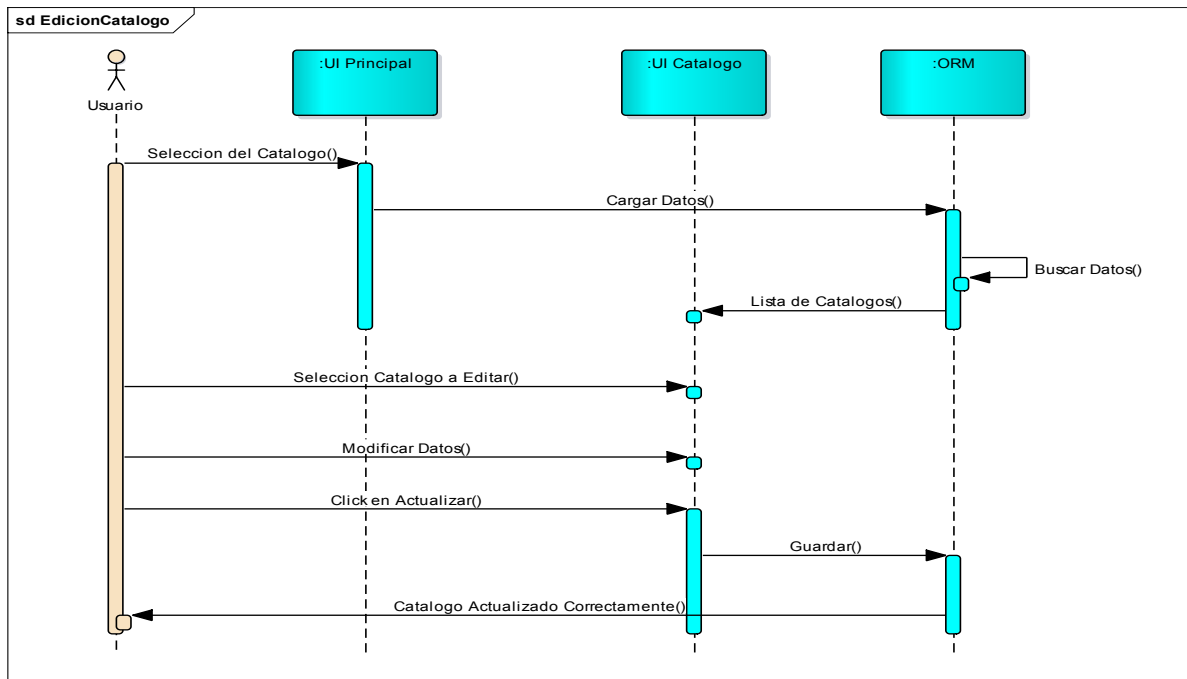


Ilustración 19, Diagrama de secuencia edición de catálogo

#### Secuencia de Acciones:

El usuario selecciona el tipo de catálogo que desea editar (Categoría o Niveles) desde la interfaz UI Principal, automáticamente se manda a llamar la interfaz que corresponde al catálogo seleccionado, en seguida se realiza la invocación del método que carga la lista de ítems de catálogos.

El usuario selecciona de la lista, el ítem a editar, modifica los datos y presiona el botón actualizar desde la interfaz UI Catalogo. Se envían los datos con los nuevos valores del ítem de catálogo a la base de datos, luego se envía un mensaje de que el Ítem de Catálogo se ha Actualizado Correctamente.



### 5.2.8.8 Diagramas de Secuencia del Servidor de Reportes

### 5.2.8.9 Generación de Reporte

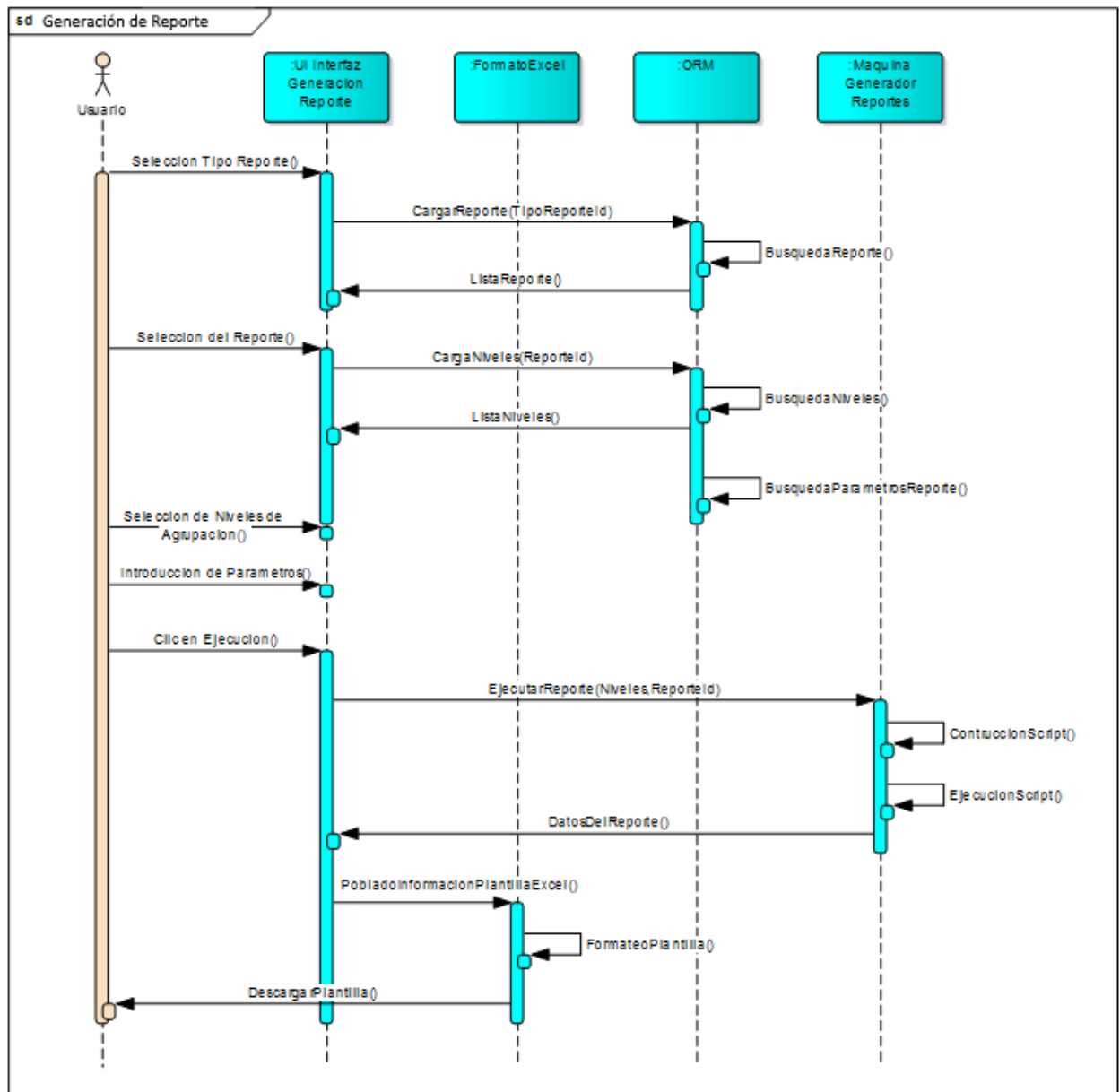


Ilustración 20: Diagrama de secuencia de la operación Generar Reporte.

### **Secuencia de Acciones:**

El usuario selecciona el tipo de reporte a generar desde la interfaz UI Generación Reporte, a continuación se lleva a cabo la invocación del método que carga la lista de reportes en base al tipo de reporte seleccionado.

El usuario selecciona el reporte que desea generar, a continuación se invoca el método que carga los niveles que están incluidos en el reporte y los niveles que fueron definidos como predeterminado con anterioridad, así como también obtiene la lista de parámetros que necesita el reporte para ejecutarse.

El usuario selecciona los niveles por los cuales se agrupará el reporte y define el orden de éstos, asimismo introduce los valores que corresponden a los parámetros del reporte. Activa el botón Ejecutar, y en seguida se envían los parámetros y el reporte seleccionados a la Máquina Generadora de Reportes en la cual se inicia el proceso de creación del código SQL necesario para obtener los datos del reporte. Una vez creado, se ejecuta el script, el cual retorna la información que corresponden a los datos del reporte.

Con ayuda del SyncFusion XlsIO se crea instancia de la plantilla en Excel y se puebla con la información obtenida con anterioridad, se realiza el formateo de la plantilla y se envía la plantilla al usuario.

### **5.2.9 Arquitectura Ejecutable**

El RUP como método se orienta a identificar los riesgos del proyecto de desarrollo de software desde la fase de elaboración a fin de eliminar o reducir aquellos riesgos principales que pudieran hacer inviable el proyecto posteriormente. El principal elemento de riesgo es la Arquitectura del Sistema, por lo cual se recomienda normalmente realizar un prototipo funcional u operativo<sup>12</sup> de la

---

<sup>12</sup> El prototipo es un modelo del comportamiento del sistema que puede ser usado para comprender o validar su funcionamiento total o parcial. El prototipo funcional u operativo implementa ciertas funcionalidades para verificar la viabilidad de su implementación. (Reyes, Ruiz, & Vivanco, 2009)

arquitectura con el propósito de verificar en la realidad el funcionamiento de los elementos más complejos de la misma, reduciendo así el riesgo del proyecto. A este prototipo se le denomina Arquitectura ejecutable.

En el caso de este proyecto, la parte de la arquitectura más crítica consiste en poder generar los datos con los valores agregados para cada nivel adecuadamente. Para abordar esta tarea, lo primero que se realizó fue el prototipo arquitectural basado en un script de generación de código SQL que se ejecutara posteriormente vía un procedimiento almacenado del tipo *pass-through*<sup>13</sup> a fin de obtener los datos del reporte listos en su orden correcto solo para ser formateado en Excel.

La construcción de dicho script de generación de código partió de una clara descripción del problema a abordar: obtener los registros de diferentes niveles de agregación anidados en el orden correcto, para lo cual se determinó que era imprescindible definir una codificación de orden que nos permitiera que aunque los datos se obtuvieran mediante diferentes consultas, al unirlos, éstos quedarán en el orden correspondiente. A continuación se describe el concepto de dicha codificación mediante un ejemplo de organización de un reporte con tres niveles:

Nivel	Descripción	Codificación de orden
0	<b>TOTAL GENERAL</b>	<b>N0</b>
1	<b>Línea 1 del Nivel 1</b>	<b>N1L1</b>
2	Línea 1 del Nivel 2 descendiente de Línea 1 del Nivel 1	N1L1-N2L1
3	Línea 1 del Nivel 3 descendiente de Línea 1 del Nivel 2	N1L1-N2L1-N3L1
3	Línea 2 del Nivel 3 descendiente de Línea 1 del Nivel 2	N1L1-N2L1-N3L2
3	Línea 3 del Nivel 3 descendiente de Línea 1 del Nivel 2	N1L1-N2L1-N3L3
2	Línea 2 del Nivel 2 descendiente de Línea 1 del Nivel 1	N1L1-N2L2
3	Línea 4 del Nivel 3 descendiente de Línea 2 del Nivel 2	N1L1-N2L2-N3L4
1	<b>Línea 2 del Nivel 1</b>	<b>N1L2</b>
2	Línea 3 del Nivel 2 descendiente de Línea 2 del Nivel 1	N1L2-N2L3
3	Línea 5 del Nivel 3 descendiente de Línea 3 del Nivel 2	N1L2-N2L3-N3L5

<sup>13</sup> Un procedimiento almacenado del tipo *pass through* o de paso, es aquel que no contiene definido el cuerpo del procedimiento internamente, solo constituye una facilidad de ejecución ante el motor de datos, recibe lo que se quiere ejecutar como parámetro en una cadena de texto, lo procesa y ejecuta y retorna los resultados.

Si se analiza la codificación de orden, se evidencia que aunque las filas hayan venido en distinto orden de diferentes consultas a la base de datos, si se realiza una unión de dichas consultas y se ordena por este código, las filas siempre quedarán en el orden correcto de jerarquía.

El siguiente paso en la construcción del prototipo fue crear un script de generación de código que implementara una consulta de agregación (con cláusula GROUP BY) por cada nivel, creando en cada consulta un campo calculado correspondiente al código de orden, construyéndolo por la concatenación del número del nivel más el Código del dato del nivel correspondiente. Finalmente las consultas se consolidan mediante la cláusula UNION. El código SQL a continuación muestra parte de este script prototipo:

```
-- Total
SELECT 0 AS Nivel, '0000000.0000000.0000000-0000000-0000000.0000000' AS Llave,
        '' AS Codigo, 'Total' AS Descripcion,
        '' AS Siglas,
        SUM(Propios) AS Propios, SUM(Tesoro) AS Tesoro,
        SUM(Donacion) AS Donacion, SUM(Prestamo) AS Prestamo,
        SUM(MtoAnual) AS Total
FROM dbo.xdcRep01(7, 2, 2015)

UNION

-- Institución
SELECT 1 AS Nivel, RIGHT ('0000000' + CAST (CodInstitucion AS varchar(7)),7) +
        '.0000000.0000000-0000000-0000000.0000000' AS Llave,
        CONVERT(varchar, CodInstitucion) AS Codigo
        , DesInstitucion AS Descripcion, Siglas, SUM(Propios) AS Propios, SUM(Tesoro) AS Tesoro
        , SUM(Donacion) AS Donacion, SUM(Prestamo) AS Prestamo, SUM(MtoAnual) AS MtoAnual
FROM dbo.xdcRep01(7, 2, 2015)
GROUP BY CodInstitucion, DesInstitucion, Siglas

UNION

-- Proyecto
SELECT 2 AS Nivel, RIGHT('0000000' + CAST(CodInstitucion AS varchar(7)),7)+ '.'+
        RIGHT('0000000' + CAST(CodProyecto AS varchar(7)),7) + '.0000000-0000000-0000000.0000000' AS Llave
        , CONVERT(varchar, CodProyecto) AS Codigo
        , DesProyecto AS Descripcion, Siglas, SUM(Propios) AS Propios, SUM(Tesoro) AS Tesoro
        , SUM(Donacion) AS Donacion, SUM(Prestamo) AS Prestamo, SUM(MtoAnual) AS MtoAnual
FROM dbo.xdcRep01(7, 2, 2014)
GROUP BY CodInstitucion, DesInstitucion, Siglas, CodProyecto, DesProyecto

ORDER BY Llave;
```

## 5.3 Fase de Construcción

### 5.3.1 Construcción de la Herramienta Diseñadora de Reportes

La herramienta Diseñadora de Reportes tiene por objetivo facilitar al usuario la elaboración y publicación de nuevos reportes, proporcionando una interfaz gráfica que permite seleccionar la función base SQL del reporte y a partir de ésta, los atributos que conformarán los niveles, la estructura del esquema de columnas del reporte, y los parámetros de filtrado.

Esta herramienta, según los requerimientos del usuario, debe ser una aplicación de escritorio, desarrollada bajo el paradigma de orientación a objetos, y deberá ser desarrollada utilizando Visual Studio 2010 como IDE y C# como lenguaje de programación. La Ilustración 21 muestra la estructura del Proyecto en Visual Studio.

Siguiendo la normativa interna de uso de las mejores prácticas de la industria, de la Dirección de Desarrollo del Banco de Proyectos (instancia de la DGIP que tiene a su cargo el desarrollo del software del SNIP), fue requerido emplear una arquitectura de múltiples capas para las aplicaciones, separando las responsabilidades de los servicios de datos (capa de acceso a datos o DAL), procesamiento de reglas del negocio (capa de lógica de negocio o BLL) e interacción con el usuario (capa de presentación PL o GUI).

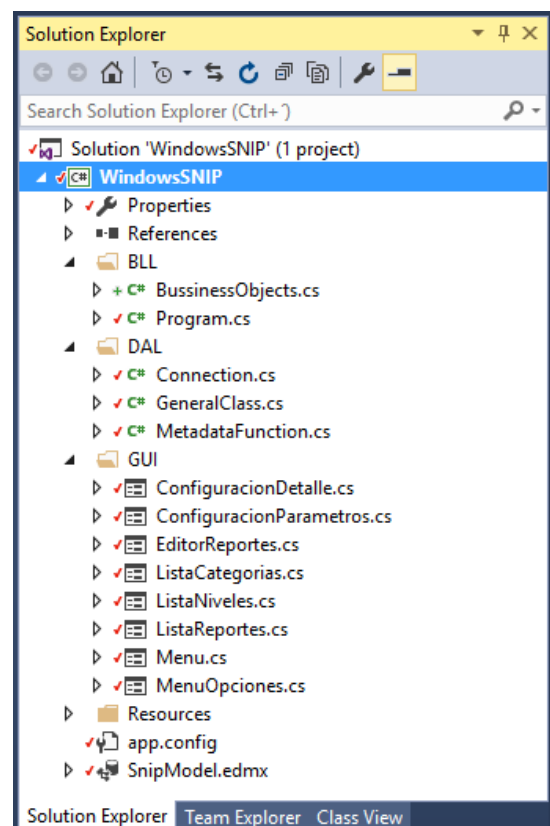
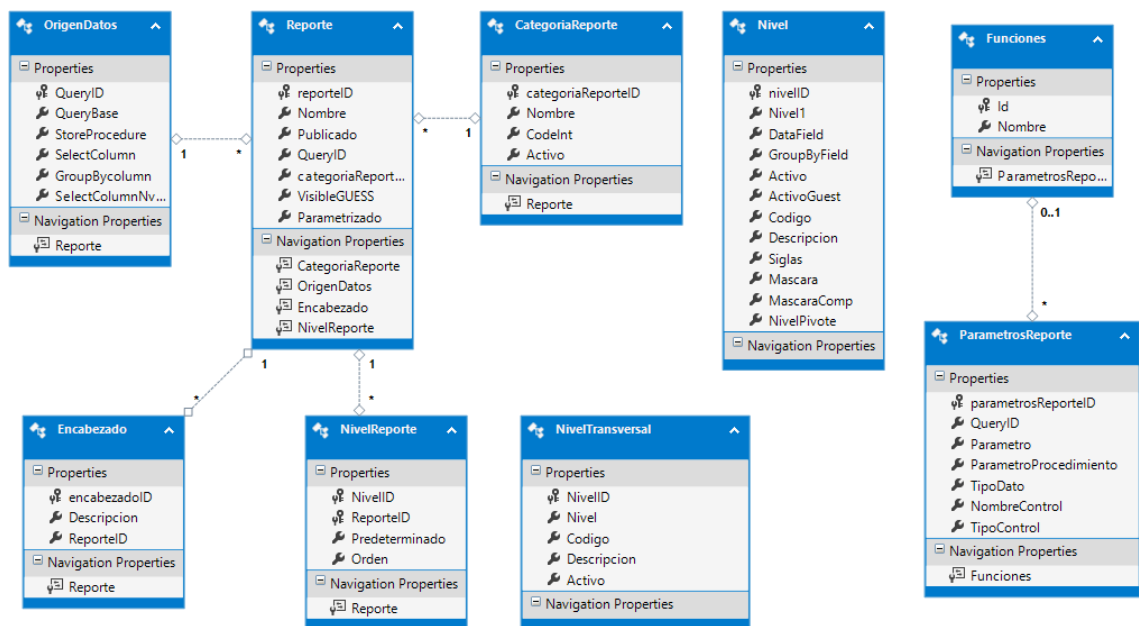


Ilustración 21: Estructura del Proyecto Diseñador de Reportes

Para facilitar la creación de la capa de acceso a datos se decidió auxiliarse de un framework de ORM, en este caso el *Entity Framework* que es uno de los componentes de .Net. La capa de acceso a datos ayuda a mantener el diseño de la base de datos separado del diseño del dominio, lo cual hace que nuestras aplicaciones sean más simples de mantener y escalar. Asimismo, automatiza las operaciones CRUD (Create, Read, Update, Delete - Crear, Leer, Actualizar, Borrar) de los datos. La figura a continuación muestra el modelo de datos base (SnipModel.edmx) empleado en el Entity Framework para la generación de la capa de acceso a datos de la herramienta Diseñadora de Reportes.



**Ilustración 22: Modelo de Datos del Entity Framework para la Herramienta Diseñadora de Reportes.**

Se utilizó Entity Framework con el enfoque *Database First*, por las bondades que este enfoque brinda para aplicaciones que parten de un modelo de datos, como es el caso de este proyecto. Permite mapear todas las tablas, vistas, procedimientos almacenados y columnas, en clases, métodos y propiedades.

El procesamiento del acceso a datos fue realizado por Entity Framework en conjunto con LINQ to Entities, que proporciona la capacidad de realizar consultas

LINQ (Language-Integrated Query) contra el modelo conceptual de Entity Framework sin tener que usar un lenguaje de consulta independiente como SQL.

Se puede decir que en el caso de Herramienta diseñadora de reportes se basó en el esquema de barra de menú gráfico introducido por Outlook (Outlook 2000) ya que la mayoría de los usuarios se encuentra familiarizado con el entorno gráfico de este software y este tipo de diseño muestra los datos de una manera fácil e intuitiva que mejora la experiencia final del usuario.

#### **5.3.1.1 Obtención de la información de la función desde SQL Server**

La fuente de datos de los reportes son funciones SQL con valores de tabla. Al momento de diseñar un nuevo reporte, la aplicación se debe encargar de presentar al usuario todas las “funciones base” disponibles en la Base de Datos para que éste seleccione la que requiere. Asimismo, una vez seleccionada la función base, se deben mostrar todos campos que aplican como niveles de agrupación, y como columnas de datos. Para lograr este cometido se requieren dos requisitos:

1. Poder acceder a la metadata de los objetos de la Base de Datos.
2. Poder identificar los diferentes objetos de la Base de Datos que son funciones base, y de cada una de éstas, los que son niveles y columnas.

Para obtener la información de metadata de los objetos de la Base de Datos se hace uso de las tablas del sistema del gestor de base de datos MS SQL Server, a través de vistas específicas que provee el gestor.

A fin de poder identificar cuales funciones son base de reportes, y los objetos de las mismas que funcionan como niveles y como columnas, se definió una nomenclatura compuesta de reglas de nombramiento basada en prefijos:

Para el caso de la función base, esta debe iniciar su nombre con el prefijo **FTRep**.

Los campos o atributos que conforman los niveles o variables del reporte deben cumplir con la siguiente nomenclatura: cada nivel se compone por dos campos: Código y Descripción. Los campos deben iniciar con los prefijos **Cod** y **Des**

respectivamente y continuar con el nombre del nivel o variable. Por ejemplo, el nivel Proyecto debe tener en la función dos campos: **CodProyecto** y **DesProyecto**. Los campos con valores numéricos son considerados columnas de datos.

El código SQL a continuación permite obtener un listado de las funciones base para reportes que existen en la Base de Datos:

```
SELECT o.object_id AS FunctionID, o.name AS FunctionName
FROM SYS.objects AS o
WHERE o.type_desc LIKE '%FUNCTION%' AND o.type IN ('TF', 'IF')
      AND o.name LIKE '%FTRep%'
      AND o.name NOT IN (SELECT QueryBase from OrigenDatos)
```

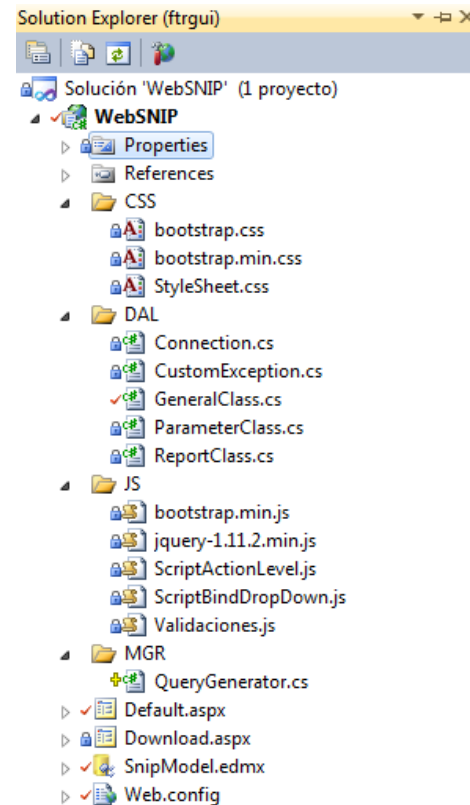
La cadena de conexión a la base de datos se almacena, según los estándares de .Net en el archivo denominado app.config. El app.config es un archivo de configuración de aplicaciones el cual se encuentra en formato xml, que contiene la información de las configuraciones de la aplicación, lo que se aprovecha para almacenar y poder manejar en el futuro cambio de servidor y base de Datos modificando la cadena de manera sencilla.

### **5.3.2 Construcción del Servidor de Reportes**

El servidor de reportes se basa en una aplicación web que provee un entorno gráfico que se le presentara al usuario final, en el cual podrá seleccionar, parametrizar y generar un reporte determinado. Permite al usuario seleccionar un tipo de reporte, un esquema de reportes, los grupos o niveles del reporte y parámetros de filtro de los datos, y una vez seleccionado el reporte, mandarlo a ejecutar. El servidor de reportes invoca a la Máquina Generadora de Reportes que es quien genera la información, que se recibe en el servidor de reportes y se formatea para presentarla al usuario final en formato Excel.



Según los requerimientos del usuario, este software debió ser implementado como una aplicación web del tipo ASP .Net Web Forms, desarrollada utilizando visual Studio 2010 como IDE y C# como lenguaje de programación. Se emplean las mismas reglas y recomendaciones de mejores prácticas que en la herramienta de diseño de reportes, utilizando Entity Framework y LINQ to Entities para la creación de la capa de acceso a datos. La Ilustración 23 muestra la estructura del proyecto del Servidor de Reportes. La carpeta MGR en esta estructura corresponde a la clase de la Máquina Generadora de Reportes. La Ilustración 24 muestra las clases del generador de reportes. En esta se puede apreciar la clase QueryGenerator que corresponde a la Máquina Generadora.



**Ilustración 23: Estructura del Proyecto del Servidor de Reportes.**

Al igual que el proyecto anterior se utiliza el archivo de configuración para almacenar la cadena de conexión a nuestra base de datos, excepto que por ser una aplicación web. Este archivo se denomina Web.config.

Para el diseño del Front-end de la aplicación, y con el propósito de mejorar la experiencia del usuario, se utilizaron diferentes marcos de trabajo o frameworks de JavaScript, principalmente BootStrap, debido a su popularidad y robustez para el diseño de sitios web, además de su compatibilidad con la mayoría de los navegadores web.

Asimismo se emplea el marco de trabajo jQuery para la interacción con la página, el manejo de los eventos y la manipulación del DOM. Este incluye código

JavaScript compatible con todos los navegadores, así también ofrece una infraestructura que aporta mucha facilidad para la creación de aplicaciones complejas del lado del cliente. Es un producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización del framework. La carpeta JS contiene estos marcos de trabajo.

La estructura del servidor de reportes es sencilla, presenta una serie de elementos de selección como listas desplegables a través de la cual el usuario seleccionará el reporte y podrá elegir o digitar los parámetros para generar el reporte.

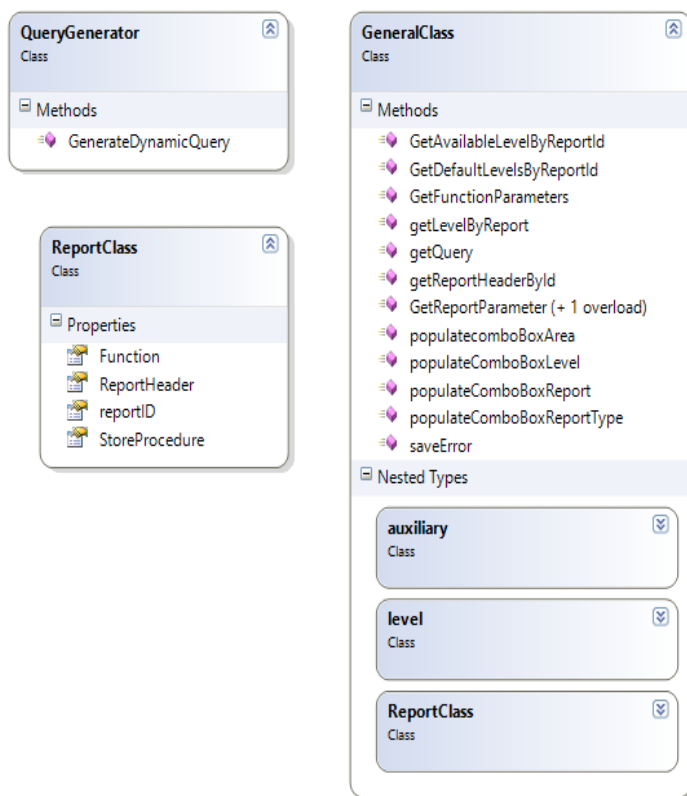


Ilustración 24: Clases del Generador de Reportes.

El servidor de reportes se basó en el diseño definido por DGIP para todas las aplicaciones que integran la reingeniería. Se parte de una página master consistente en un esquema en L formado por una cabecera conformada por un banner y un menú izquierdo. Se puede mencionar que se basó la app en el conjunto de estándares de diseño que se conoce como Web 2: Diseño orientado al usuario, interactividad y facilidad de

uso, Ajax; así como las nuevas tendencias orientadas a responsividad son HTML5 con CSS3: Bootstrap, JQuery

### **5.3.3 Construcción de la Máquina Generadora de Reportes**

Su función es la creación del script necesario para la generación del reporte solicitado desde la interfaz web. Parte de la definición del reporte almacenada en la base o área de datos del Generador de Reportes y construye el script SQL para ejecutarlo a través de un procedimiento almacenado tipo *pass through*.

#### ***5.3.3.1 Proceso de construcción del SCRIPT***

El proceso de construcción del script que obtiene los datos del reporte cuenta con ciertas particularidades que se describen a continuación:

1. El script está formado por uniones sucesivas de consultas de datos de los diferentes niveles de agrupación utilizando como origen de datos la función con valores de tabla base del reporte para todas ellas.
2. Cada registro cuenta con una llave especial que está estructurada por una máscara que ha sido previamente definida. Esta llave se crea de manera dinámica y contiene los niveles de agrupación seleccionados más los códigos de los niveles (Cada nivel está definido por un par <Código, Descripción>). Esta llave es de vital importancia debido a que es la encargada de efectuar el ordenamiento de la información, es decir se encarga de mostrar los datos en el orden solicitado por el usuario. Se han definido funciones escalares que tienen la finalidad de construir cada parte de la llave para la consulta SQL.
3. El proceso de construcción del script recorre o itera a través de los niveles que le han sido solicitados y en conjunto con las funciones escalares genera la llave de ordenamiento, la cláusula SELECT que comprende las columnas seleccionadas que se mostraran en el reporte y la cláusula

GROUP BY con las columnas por las cuales se está agrupando cada instrucción.

4. Existe un nivel cero o nivel predeterminado que corresponde al **Total General** del reporte, que está formado por los valores totalizados. Su llave tiene una construcción predefinida con todos los campos en cero a fin que siempre esté encabezando el reporte.

El resto de niveles de agrupación presentan las siguientes particularidades:

- a) Su definición cuenta con un par **<key, value>** definido como **<Código, Descripción>**. Estos campos son identificados en la función debido a la nomenclatura definida para ello que consiste en que sus nombres deben empezar con los prefijos **Cod** y **Des** respectivamente.
- b) La máscara para cada nivel es estándar, es un numero de 6 dígitos rellenado con ceros (0000000)

A continuación se muestra la parte nuclear del código para implementar el algoritmo de creación de llaves únicas de los registros para asegurar el orden correcto de los datos.

```

--ARMANDO LA QUERY INICIAL CORRESPONDIENTE AL NIVEL 0
SET @query = 'SELECT 0 AS Nivel,' + (SELECT dbo.Fn_Generate_Key_V2(@niveles,0)) +
' as Llave, '' as Codigo, ''Total'' as Descripción, '' as Siglas,' + @select +
' FROM ' + @from

--ITERANDO A TRAVES DE LOS NIVELES
while (@cont <= (select COUNT(*) from Split(@niveles',')))
begin
    set @query = @Query + CHAR(13) + ' UNION ' + CHAR(13) + 'SELECT ' +
    CONVERT(varchar,@cont) + ' as Nivel, '
    set @query = @query + (SELECT dbo.Fn_Generate_Key_V2(@niveles,@cont)) + ' as
    Llave, ' + (select dbo.Fn_Generate_Select(@niveles,@cont,@reportID))
    set @query = @query + ' FROM ' + @from + ' GROUP BY '
    set @query = @query + (select
    dbo.Fn_Generate_GroupBy_V2(@niveles,@cont,@reportID))
    set @cont = @cont + 1
end

set @query = @query + CHAR(13) + 'ORDER BY Llave'
print @query

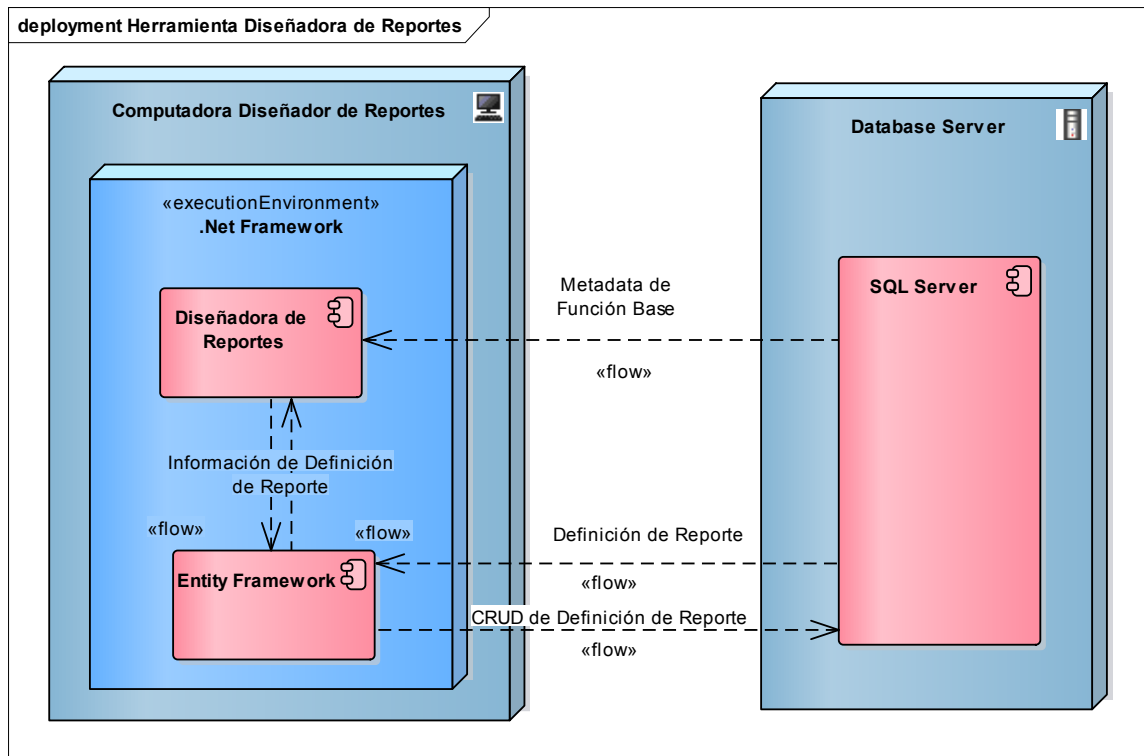
```

## 5.4 Fase de Transición

La fase de transición o despliegue corresponde al proceso de implantación del sistema y su liberación a los usuarios finales. Integra todo el proceso de instalación, inducción, pruebas de usuario, aceptación y puesta en servicio. Uno de los principales artefactos para planificar la instalación es la arquitectura de despliegue.

### 5.4.1 Arquitectura de Despliegue

El diagrama de arquitectura de despliegue del sistema es una representación gráfica y simplificada de cómo se instalará el sistema en la infraestructura tecnológica (computadoras) de cara al usuario final. Para efectos de este trabajo, que implica el desarrollo e implantación de dos aplicativos: la Herramienta Diseñadora de Reportes y el Servidor de Reportes; se diseñaron los modelos de arquitectura de despliegue de ambas.



**Ilustración 25: Arquitectura de Despliegue Herramienta Diseñadora de Reportes.**

La Ilustración 25 muestra el diagrama de la Arquitectura de Despliegue de la Herramienta Diseñadora de Reportes, que consiste en una aplicación .Net del tipo Windows Forms que se comunica con la Base de Datos del Reporte para obtener la información de las Funciones Base de Reportes y crear, clonar, actualizar o borrar las definiciones de Reportes según lo requiera el usuario final.

La Ilustración 26 muestra el diagrama de Arquitectura de despliegue de la aplicación web del Servidor de Reportes.

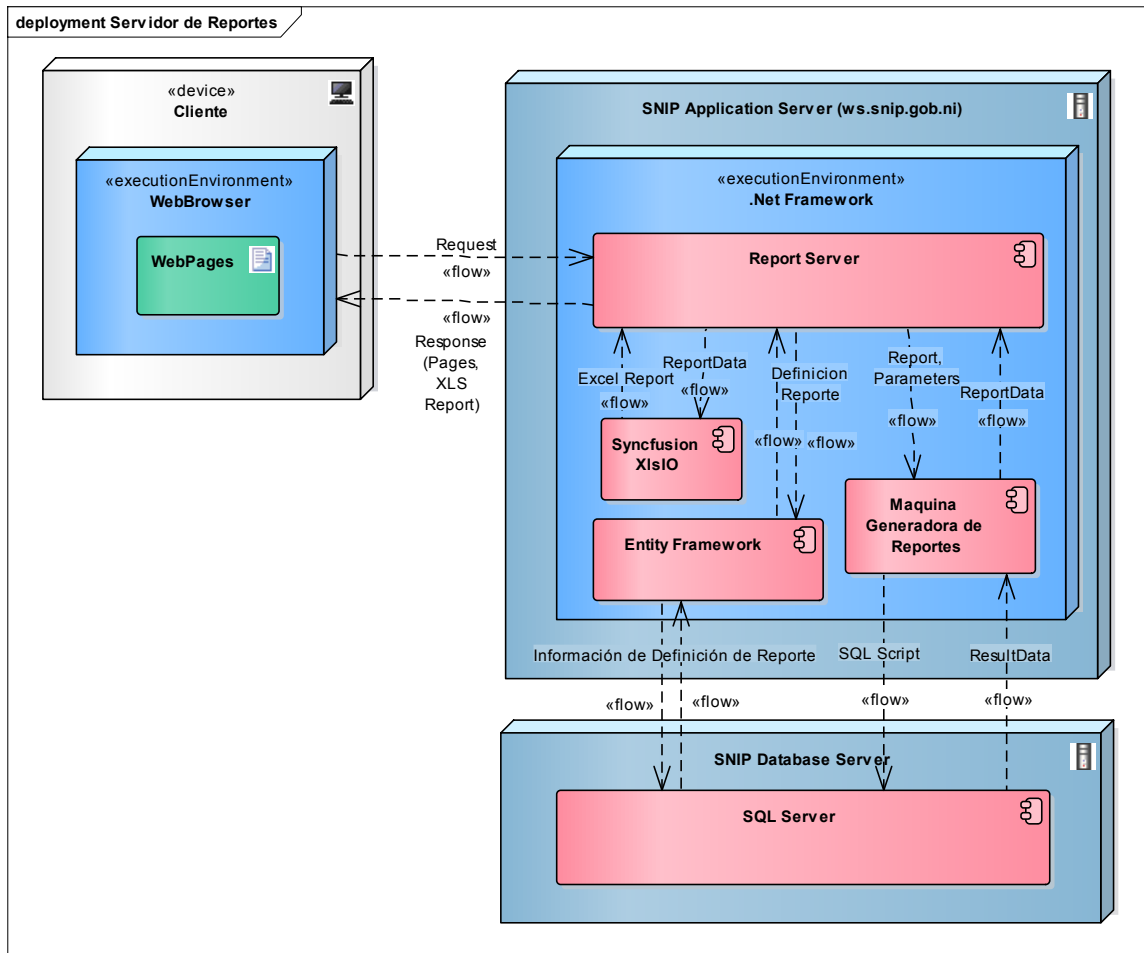


Ilustración 26. Diagrama de Arquitectura de Despliegue del Servidor de Reportes.

El diagrama muestra la interrelación y flujo de información entre los diferentes marcos de trabajo y componentes de la aplicación, así como su despliegue en dos servidores: El servidor de aplicaciones y el de base de datos del SNIP.

El cliente selecciona un reporte y sus parámetros a través de las páginas web que presenta el servidor de reportes (que construye a partir de la información obtenida de la base de datos a través del Entity Framework). El Servidor de Reportes genera los datos del reporte (a través de la Máquina Generadora que se conecta directamente a la base de datos), lo transforma en formato Excel (con auxilio del Syncfusion XlsIO) y lo sirve al cliente como un flujo de bytes que el usuario final puede abrir directamente con Excel.

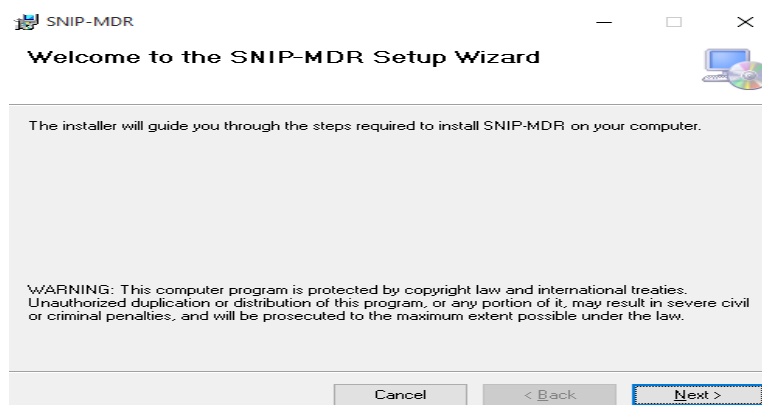
## 5.4.2 Proceso de Instalación

### 5.4.2.1 *Proceso de instalación de la herramienta diseñadora de reportes*

Para el proceso de instalación se debe de contar con los archivos ejecutables que se han generado para el proceso de instalación. Los nombres de los archivos son snip-mdre.msi y setup.exe. El primero, más compacto, requiere de soporte del instalador de Windows y el segundo es totalmente autocontenido.

El proceso de instalación es sencillo y cuenta simplemente con 5 pasos:

1. El primero consiste en ejecutar el archivo de instalación correspondiente a extensión msi que muestra un mensaje de ayuda para el proceso de



**Ilustración 27: Instalación de SNIP-MDR**

instalación. Para continuar se debe pulsar el botón [Next].



- Seguido de leer la advertencia se procede a la selección de la ruta de ubicación donde se instalara el programa la ruta por defecto es `%PROGRAMFILES(X86)%\SNIP\SNIP-MDR`. También se puede consultar el costo del espacio que consume el programa (1308 kb) y si estará disponible para todos los usuarios o solo para el administrador.

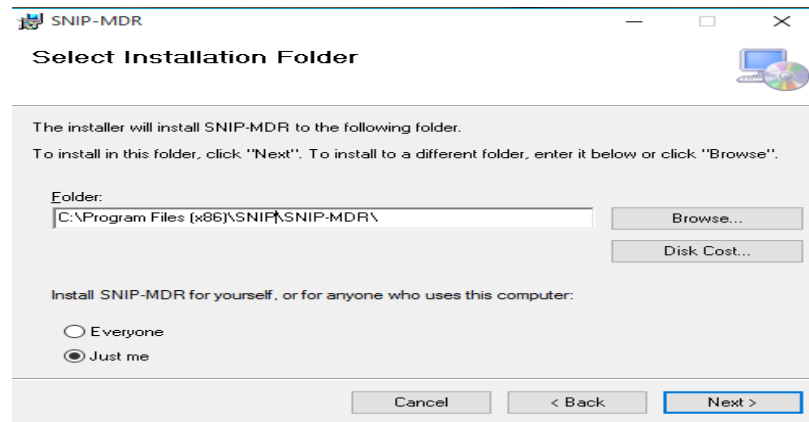


Ilustración 28, Instalación SNIP-MDR

Para continuar se debe pulsar el botón [Next].

- Finalmente el mensaje de confirmación del programa, se confirma pulsando el botón [Next].

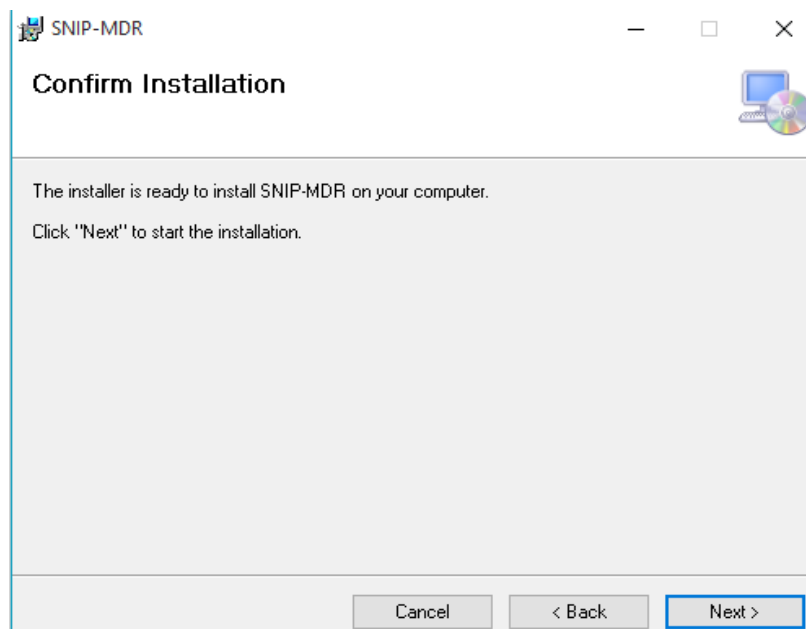
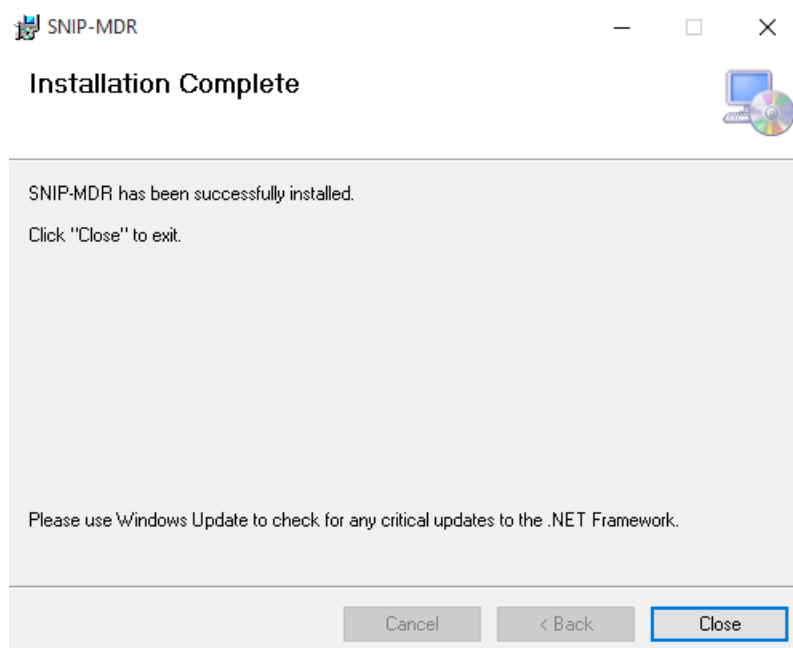


Ilustración 29, Instalación SNIP-MDR

4. Se cierra el programa de instalación, se comprueba en la PC si está

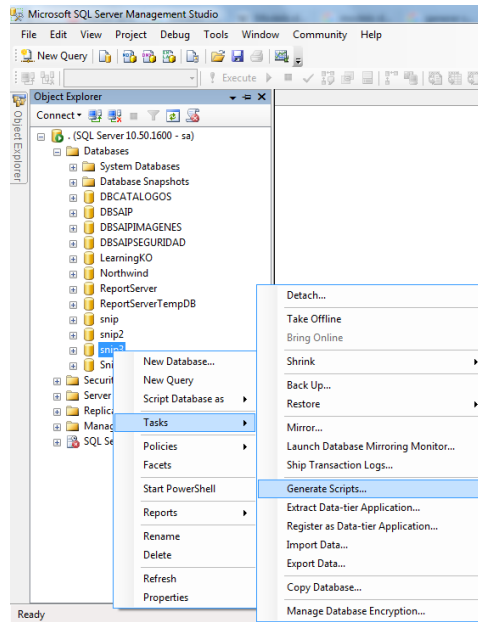


**Ilustración 30: Instalación completa SNIP-MDR**

instalado correctamente y funciona correctamente. Pulse el botón [Close] para finalizar.

#### ***5.4.2.2 Proceso de instalación del Servidor de reportes***

Para la instalación del servidor de reportes se requiere instalar la base de datos del Generador en un esquema o área de la base de datos en producción, o en una base de datos separada. En el caso del despliegue en el SNIP, el cliente decidió un enfoque de bases de datos separada. Para sistematizar la instalación se creó un script de exportación e importación que permite trasladar tanto la estructura como los datos de la base de datos. La generación del script se realizó en base a un asistente que provee SQL Server. La Ilustración 31 muestra la localización de este asistente en SQL Server.



**Ilustración 31: Funcionalidad de SQL para exportar la base de datos.**

El script generado se ejecutó en el servidor de base de datos del SNIP creando una base de datos separada para el generador de reportes. Se creó un usuario específico para que la aplicación tuviera acceso a las bases de datos, tanto la de reportes como la base operativa. En el Anexo 4. Se adjunta una guía para la creación del script de exportación de base de datos según los pasos realizados para el presente proyecto. Este podría facilitar futuras instalaciones en otros ambientes.

Una vez instalada la base de datos se procedió a instalar la aplicación en el servidor web. Desde el Visual Studio se publicó la aplicación, con lo cual se generaron paquetes, los que fueron desplegados en el servidor web en un nuevo contenedor de aplicación web.

Se realizaron las pruebas de funcionamiento y se cerró el procedimiento exitosamente.

#### **5.4.2.3 Acceso al Servidor de Reportes**

El acceso al servidor de reportes será por medio de internet, para lo cual el cliente requiere tener acceso y un navegador. El URL para acceder al servidor de reportes es: <http://ws.snip.gob.ni/BPRep/>

#### **5.4.3 Inducción a usuarios**

Se realizó un proceso de inducción a usuarios para el correcto aprovechamiento del generador de reportes. En una primera fase se capacitó al usuario encargado de diseñar los reportes. En vista que este es un usuario experto, el proceso se llevó a cabo sin ningún contratiempo y el usuario rápidamente desarrolló las competencias para operar la herramienta Diseñadora de Reportes.

En un segundo momento se capacitaron 10 personas de DGIP para el uso del servidor de reportes. Gracias a que el diseño de la interfaz de usuario era muy similar a la de la herramienta actual, con los mismos conceptos y controles, no hubo, tampoco ningún problema en su adopción.

#### **5.4.4 Pruebas de usuario**

Los usuarios realizaron una serie de pruebas para validar que el servidor de reportes genera los reportes solicitados con sus parámetros, que los resultados son correctos y se corresponden a los generados por la herramienta actual. Todas las pruebas fueron satisfactorias.

Finalizadas las pruebas, el sistema se aceptó por parte del Director de Desarrollo del Banco de Proyectos y se puso en servicio a lo interno de DGIP, también se publicó al internet, pero falta integrarlo al Banco de Proyectos todavía. Este es un proceso que realizará la Dirección de Desarrollo del Banco de Proyectos según sus procedimientos internos.

#### **5.4.5 Documentación**

Además de la documentación técnica integrada en el código fuente, a solicitud del Director de Desarrollo del Banco de Proyectos, se deja el informe de la monografía como material de referencia técnica, principalmente la parte correspondiente al Proceso RUP. Adicionalmente se preparó una guía de referencia técnica para facilitar el mantenimiento.

Asimismo se elaboraron manuales de usuario para las dos herramientas: el diseñador de reportes y el servidor de reportes. Toda la documentación extra generada se adjunta como anexo.

## 6 Evaluación de resultados

Para la evaluación de los resultados de la aplicación del nuevo generador de reportes para el Banco de Proyectos del SNIP, en la Dirección General de Inversión Pública, se establecieron primeramente las métricas de evaluación con sus medidas o variables, los instrumentos y el método de evaluación para obtener sus valores (Olsina, 2012). De la aplicación del método se obtuvieron los datos, que fueron procesados para definir los resultados de la evaluación.

Para facilitar la operatividad del proceso de evaluación, éste se dividió en dos fases o subprocesos, en dependencia de las dos funcionalidades principales de la solución propuesta:

- ✓ Evaluación de la creación de un nuevo reporte
- ✓ Evaluación de la generación de reportes.

### 6.1 Métricas

Se identificaron dos tipos de métricas<sup>14</sup> a evaluar en ambos subprocesos: Desempeño y Facilidad de Uso. Para cada tipo de métrica se definieron además sus variables a fin de comprobar los beneficios del nuevo módulo generador de reportes respecto al anterior. A continuación se listan las variables por cada tipo de métrica:

**V0. Generador de Reporte:** Variable independiente que representa al generador de reportes que se está evaluando. Variable discreta con lista de valores: Anterior, Nuevo.

---

<sup>14</sup> Entiéndase como métrica en este contexto a un “conjunto de medidas o variables que caracterizan un concepto medible” (ISO 15939) y como “método y escala de medición” (ISO 14598 -1:1999).

- A. **Desempeño:** Para evaluar el concepto de desempeño se empleará la métrica de latencia del sistema a través de la variable tiempo de respuesta, y las comparaciones de desempeño como indicador (razón) y como ventaja (porcentaje):

**V1. Tiempo de respuesta (t):** Tiempo requerido para generar un reporte determinado. Variable numérica real. Unidad de medida: segundos (s).

**V2. Razón de Rendimiento (n):** Cociente entre el Tiempo de Respuesta del Generador Anterior sobre el Tiempo de Respuesta del Generador Nuevo. Esta métrica permite la comparación de rendimientos entre los generadores: Si es  $> 1$ , entonces el nuevo generador es mejor que el actual. Si  $= 1$ , son idénticos en rendimiento y si  $< 1$ , el nuevo generador es inferior al actual en cuanto a rendimiento.

**V3. Mejora de rendimiento neta porcentual (n%):** Se define como  $n-1$  multiplicado por 100%. Expresa en qué medida el nuevo generador es mejor que el otro.

A continuación se definen las ecuaciones de estas variables. Nótese que para los efectos de este proyecto, se empleará la segunda expresión en el cálculo de  $n$ .

$$n = \frac{R_{nuevo}}{R_{anterior}} = \frac{t_{anterior}}{t_{nuevo}}$$

Ecuación 1: Razón de Rendimiento

$$n_{\%} = (n - 1) * 100\%$$

Ecuación 2: Ventaja neta porcentual de rendimiento

- B. **Facilidad de uso:** Para evaluar el concepto de facilidad de uso se tomaron en cuenta dos métricas, la primera en dependencia del esfuerzo del usuario y la segunda en cuanto a la simplicidad de interacción con el sistema:

**V4. Pasos de procedimiento:** Cantidad de pasos de procedimiento necesarios para lograr una tarea terminada. Variable numérica entera. Unidad de medida: adimensional.

**V5. Simplicidad:** Grado de simplificación y adecuación del diseño de la interfaz que facilita entender cómo interactuar con ella para lograr instruirle al sistema lo que se quiere. Variable discreta con categoría o lista de valores: Simple, Complejo.

## **6.2 Método de evaluación**

Para la evaluación de las métricas de Desempeño se seleccionó el método de testing o pruebas.

Para la evaluación de las métricas de Facilidad de Uso, se seleccionó el método de consulta (Olsina, 2012).

Se definió la carga de trabajo que consistió en tres (3) de los reportes más usados en el devenir cotidiano de DGIP:

1. Reporte financiero de Programación anual por tipo de recurso
2. Reporte Comparativo financiero de Programación anual vs ejecución
3. Reporte de Ejecución de Indicadores Físicos

Se definieron los evaluadores o testers para cada subproceso. En el caso del subproceso de Evaluación de la creación de un nuevo reporte, se seleccionó un único evaluador, que es quien diseña los reportes en el Banco de Proyectos con la herramienta actual. Este evaluador es un (1) usuario experto, ingeniero en computación, con dominio conceptual del negocio.

En el caso de subproceso de Evaluación de la generación de reportes, el equipo evaluador fue conformado por diez (10) especialistas de inversión pública que atienden a todas las instituciones del SNIP y que son usuarios expertos de la máquina de generación de reportes actual.



### **6.2.1 Procedimiento para evaluación de la creación de un reporte**

- Se definió el set de pruebas consistente en la creación o diseño de cada uno de los reportes de la carga de trabajo con cada una de las herramientas (primero con la anterior y posteriormente con la nueva).
- Se llevaron a cabo las pruebas del set tomando, de cada una de ellas, mediciones de las variables: Tiempo de respuesta (empleando un cronómetro), Pasos de procedimiento (por conteo) y Simplicidad (mediante percepción de usuario). El registro de estos datos se realizó por parte del tester en un formato o ficha de resultados.
- Se tabularon los datos obtenidos de las mediciones en hoja de cálculo de Excel y se analizaron para poder concluir los resultados de la evaluación.

### **6.2.2 Procedimiento para evaluación de la generación de un reporte**

- Se definió el set de pruebas consistente en la generación de cada uno de los reportes de la carga de trabajo con cada una de las herramientas (primero con la anterior y posteriormente con la nueva).
- Se llevaron a cabo las pruebas del set tomando, de cada una de ellas, mediciones de las variables: Tiempo de respuesta (empleando un cronómetro), Pasos de procedimiento (por conteo) y Simplicidad (mediante percepción del tester). El registro de estos datos se realizó por parte del tester en un formato o ficha de resultados.
- Se tabularon los datos obtenidos de las mediciones en hoja de cálculo de Excel y se analizaron para poder concluir los resultados de la evaluación.

En base a los datos obtenidos y tomando como referencia las variables anteriormente planteadas se determinó si el nuevo generador de reportes desarrollado para el Banco de Proyectos del SNIP es idóneo para cumplir los objetivos planteado y podrá proveer los beneficios esperados.

## 6.3 Resultados de la evaluación del proceso de creación de reportes.

### 6.3.1 Evaluación de Desempeño.

La tabla 1 a continuación resume los resultados de las mediciones. Para la evaluación comparativa se emplearon las métricas de razón de rendimiento  $n$  y mejora neta porcentual de rendimiento<sup>15</sup>  $n\%$  las cuales establecen en qué medida un sistema aventaja al otro. En promedio, el diseño de un nuevo reporte se completa en 67 minutos (4,000 segundos) en el diseñador anterior y tan solo en 46 minutos (2,760 segundos) en el nuevo. Los resultados evidencian que el nuevo diseñador de reportes tiene mejor desempeño que el anterior de acuerdo a la latencia del proceso de diseño de nuevo reporte, pues su razón de rendimiento es 1.5, teniendo una ventaja de rendimiento del 45% sobre el anterior

Tabla 1: Resultado latencia de Creación de Reportes.

Caso de prueba	Generador Anterior	Generador Nuevo	n	n%
	t [s]	t [s]		
Test 1	2,400	1,800	1.3	33%
Test 2	5,820	4,140	1.4	41%
Test 3	3,780	2,340	1.6	62%
Media	<b>4,000</b>	<b>2,760</b>	<b>1.5</b>	<b>45%</b>

---

<sup>15</sup> La definición de estas métricas y su método de cálculo fue definido en la sección de Metodología de la Evaluación. Su sustento científico - técnico se puede verificar en las referencias: (Hennessy & Patterson, 2012) y (Díaz, 2015)

### **6.3.2 Evaluación de Facilidad de uso**

Para esta evaluación se tomaron en cuenta dos medidas, la primera en dependencia del esfuerzo requerido al usuario para su uso (Pasos de procedimiento) y la segunda en cuanto a la simplicidad de interacción con el sistema (Simplicidad de uso).

#### **6.3.2.1 Pasos de procedimiento**

La tabla a continuación compara la cantidad de pasos de procedimiento necesarios para la creación de cada reporte. Los datos revelan que el nuevo generador requiere menos pasos para completar la tarea en todos los casos, concluyéndose que requiere 8 pasos menos que el anterior, por lo cual es más fácil de usar.

Tabla 2: Pasos de procedimiento para crear un reporte.

Caso de prueba	Generador Anterior	Generador Nuevo
	Cantidad de pasos	Cantidad de pasos
Test 1	22	14
Test 2	22	14
Test 3	22	14
Media	22	14

#### **6.3.2.2 Simplicidad de uso**

La evaluación de la simplicidad o grado de simplificación del diseño de la interfaz gráfica de la Herramienta Diseñadora de Reportes se realizó a través de la percepción del usuario. La tabla 3 muestra el resultado, en el que se evidencia que la nueva diseñadora de reportes es más simple y por tanto fácil de usar para el operador.

Tabla 3: Resultado de Simplicidad de Uso.

Caso de prueba	Generador Anterior	Generador Nuevo
	Simplicidad {Complejo, Simple}	Simplicidad {Complejo, Simple}
Test 1	Complejo	Simple
Test 2	Complejo	Simple
Test 3	Complejo	Simple
<b>Moda</b>	<b>Complejo</b>	<b>Simple</b>

En conclusión, la herramienta diseñadora de reportes del nuevo módulo de generación de reportes es más fácil de usar que la del anterior por cuanto tiene menos pasos procedimentales e interfaz de usuario más simple para la creación del reporte.

## 6.4 Evaluación del proceso de generación de reportes.

La evaluación de la generación de reportes se realizó a través de la aplicación de test o pruebas comparativas de ejecución, para las cuales la carga se conformó por los tres reportes creados en la Creación de Reportes.

### 6.4.1 Evaluación del Desempeño.

#### **Test No. 1: Generación del reporte de Programación anual por tipo de recurso**

La tabla 4 muestra los resultados de las aplicaciones del caso de prueba test No. 1. En este test en particular se quiso registrar todos los tiempos de las distintas fases del proceso de generación. En el generador anterior, una vez se seleccionan los parámetros y se pulsa el botón de Generar, inicia el proceso “en servidor” hasta que el sistema entrega un link para descargar el reporte. Una vez que el usuario sigue el link, el navegador descarga el reporte y la aplicación de formato lo transforma en un archivo de Excel mediante un proceso local en la máquina cliente. Por tanto, se registraron dos tiempos, el Tiempo 1 que inicia con

el comando generar y concluye con la recepción del link de descarga, y el Tiempo 2 que inicia con el pulsado en el link de descarga y finaliza una vez que se tiene el archivo de Excel resultante abierto y listo para su uso. Ambos tiempos se expresan en segundos.

**Tabla 4: Resultados de aplicación de Test No. 1**

TEST No 1		Generador de Reportes Anterior						Nuevo Generador de Reportes				
TESTER	Hora Inicio	Hora Link	Hora Fin	Tiempo1	Tiempo2	Tiempo	Hora Inicio	Hora Descarga	Hora Fin	Tiempo1	Tiempo2	Tiempo
1	10:39:32	10:40:28	10:40:59	56	31	87	10:42:33	10:42:45	10:43:11	12	26	38
2	11:02:30	11:03:19	11:03:39	49	20	69	10:40:00	10:40:10	10:40:19	10	09	19
3	11:17:25	11:18:01	11:18:35	36	34	70	11:06:15	11:06:24	11:06:39	09	15	24
4	1:23:35	1:24:33	1:25:08	58	35	93	1:31:50	1:32:05	1:32:49	15	44	59
5	1:28:10	1:28:49	1:29:17	39	28	67	1:32:05	1:32:16	1:32:34	11	18	29
6	3:30:02	3:30:37	3:30:59	35	22	57	3:31:05	3:31:15	3:31:17	10	02	12
7	3:33:00	3:33:40	3:34:01	40	21	61	3:34:30	3:34:39	3:34:41	09	02	11
8	3:45:04	3:45:48	3:46:15	44	27	71	3:46:30	3:46:41	3:46:43	11	02	13
9	4:00:00	4:00:39	4:01:07	39	28	67	4:01:30	4:01:42	4:01:45	12	03	15
10	4:03:05	4:03:49	4:04:19	44	30	74	4:05:00	4:05:11	4:05:13	11	02	13

En el generador nuevo, aunque el proceso es unificado, sí se pueden determinar asimismo dos tiempos: el Tiempo 1 que transcurre desde que se emite el comando Generar hasta que se inicia el proceso de descarga y el Tiempo 2 hasta que se completa la descarga. Este segundo tiempo puede ser significativo para las instituciones en dependencia de su velocidad de acceso.

Un aspecto interesante de la prueba 1 es que se quiso determinar la diferencia de los tiempos de respuesta si se genera el reporte desde el Internet o desde la red local de DGIP a fin de apreciar el impacto en la operación de las instituciones. Para esto, los primeros cinco testers se conectaron a través de Internet y el resto de ellos lo hicieron a través de la red local (LAN).

Los resultados de consolidaron en función de los tiempos parciales y el tiempo total evidenciándose que el nuevo generador de reportes es más rápido (tiene mejor desempeño) que el anterior.

El Gráfico 1 a continuación muestra la relación entre la ejecución de las pruebas con los dos generadores de reportes.

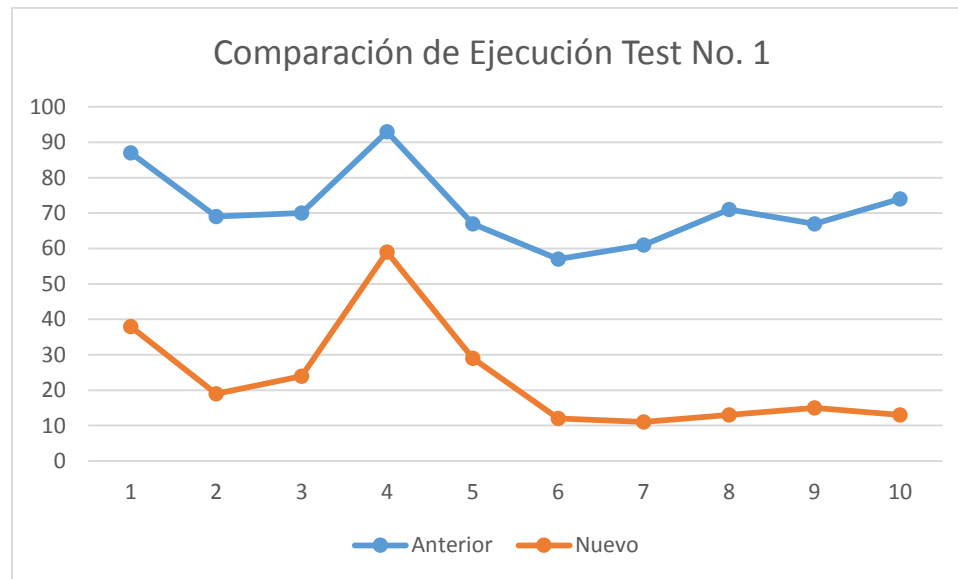


Gráfico 1: Ejecución Test No 1

La tabla 5 consolida los datos de las mediciones realizadas, mostrando que en promedio, el reporte tarda 72 segundos en generarse con el generador anterior y solo 23 segundos con el nuevo generador. Como resultado, se puede afirmar que su relación de rendimiento es 3.9, es decir, que el nuevo generador aventaja al anterior en tiempo de respuesta en un 286%, o desde otra perspectiva, que su latencia es solo el 33% de la latencia del generador anterior con lo cual se tiene un ahorro de tiempo, o aceleración de rendimiento del 67% al emplear el nuevo generador.

Se puede asimismo notar que los tiempos de respuesta se incrementan ligeramente en el caso del generador anterior si el usuario se conecta desde el Internet (17%) pero estos se incrementan considerablemente (164%) en el nuevo, lo cual depende mucho del ancho de banda que se tenga. Cabe mencionar que el enlace empleado para la prueba tenía un ancho de banda inestable (módem

inalámbrico) que es un factor que puede estar condicionando negativamente estos resultados.

Tabla 5: Evaluación comparativa de resultados de los Generadores de Reportes.

TESTER	Generador de Reportes Anterior			Nuevo Generador de Reportes			n	n%
	T1	T2	T	T1	T2	T		
1	56	31	87	12	26	38	2.3	129%
2	49	20	69	10	9	19	3.6	263%
3	36	34	70	9	15	24	2.9	192%
4	58	35	93	15	44	59	1.6	58%
5	39	28	67	11	18	29	2.3	131%
6	35	22	57	10	2	12	4.8	375%
7	40	21	61	9	2	11	5.5	455%
8	44	27	71	11	2	13	5.5	446%
9	39	28	67	12	3	15	4.5	347%
10	44	30	74	11	2	13	5.7	469%
<b>Media:</b>	<b>44</b>	<b>28</b>	<b>72</b>	<b>11</b>	<b>12</b>	<b>23</b>	<b>3.9</b>	<b>286%</b>
Media Internet	48	30	77	11	22	34	2.5	154%
Media LAN	40	26	66	11	2	13	5.2	418%
Incremento			17%				164%	

## Test No. 2: Generación del Comparativo financiero de Programación anual vs ejecución.

La tabla 6 muestra los resultados de la aplicación de la prueba o test No. 2. Los resultados para esta prueba se registraron como un todo, sin separar las partes como en la prueba anterior por no ser necesario. Al igual que en la prueba anterior los resultados muestran una ventaja de rendimiento del nuevo generador sobre el anterior.

Como se evidencia en la tabla, en promedio se registra una latencia de unos 33 segundos en obtenerse el reporte con el nuevo generador contra unos 90 segundos con el anterior. Esto da una razón de rendimiento de 2.8 o 179% de ventaja neta del nuevo generador sobre el anterior. Desde la perspectiva inversa, se puede decir que el nuevo generador tiene una latencia de solamente el 36%

respecto al tiempo o latencia del generador anterior con lo cual se tiene una mejora del 64% sobre aquel.

Tabla 6: Resultados de Test No. 2.

TEST No	2							
TESTER	Generador de Reportes Anterior			Nuevo Generador de Reportes			n	n%
	Hora Inicio	Hora Fin	Tiempo	Hora Inicio	Hora Fin	Tiempo		
1	11:31:50	11:33:24	94	11:28:45	11:29:16	31	3.0	203%
2	12:30:00	12:31:29	89	12:35:22	12:35:54	32	2.8	178%
3	12:42:15	12:43:39	84	12:50:36	12:51:04	28	3.0	200%
4	1:23:05	1:24:41	96	13:27:16	13:28:04	48	2.0	100%
5	1:40:00	1:41:27	87	1:42:55	1:43:24	29	3.0	200%
6	1:55:02	1:56:37	95	1:57:38	1:58:12	34	2.8	179%
7	2:00:04	2:01:31	87	2:05:02	2:05:31	29	3.0	200%
8	2:09:05	2:10:34	89	2:13:27	2:13:59	32	2.8	178%
9	2:17:08	2:18:39	91	2:21:04	2:21:35	31	2.9	194%
10	5:21:23	5:22:49	86	5:31:00	5:31:33	33	2.6	161%
MEDIA			90				33	179%

.La Gráfica a continuación muestra la relación entre los resultados de la ejecución del Test 2 con el generador anterior y el nuevo generador de reportes.

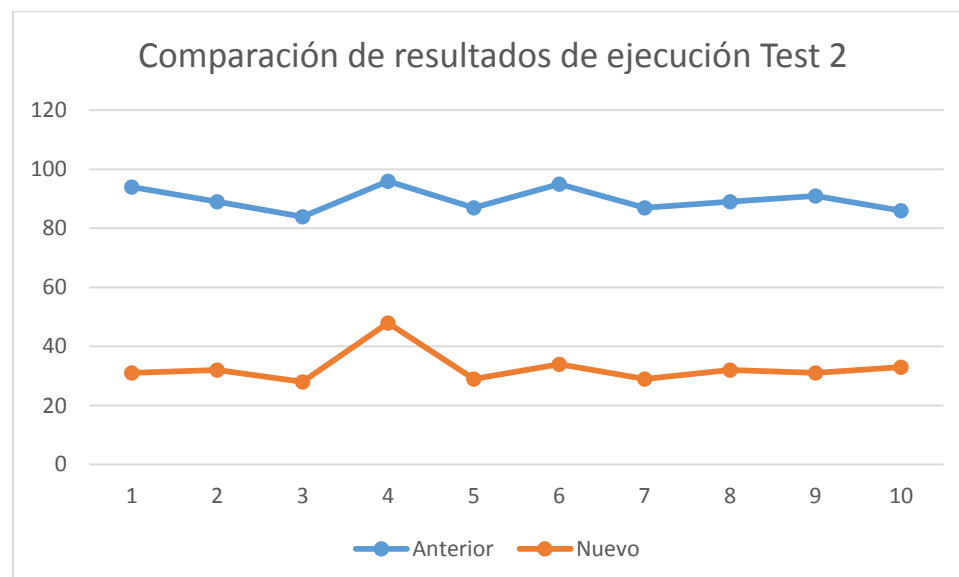


Gráfico 2: Comparación de ejecución del Test 2.



### Test No. 3: Generación del Reporte de Ejecución de Indicadores Físicos.

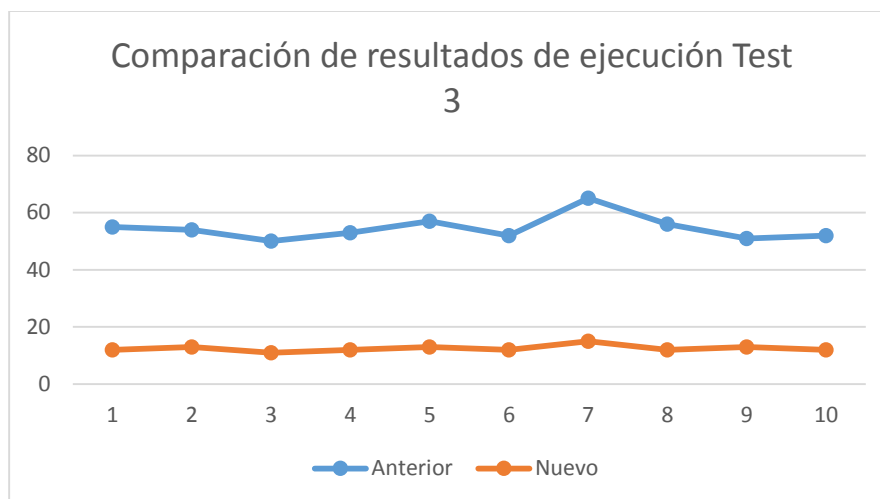
La tabla 7 muestra los resultados de la aplicación del test No. 3. Al igual que en la pruebas anteriores los resultados muestran una ventaja de rendimiento del nuevo generador sobre el anterior.

En este test se evidencia en los datos un amplio margen de mejora del nuevo generador respecto al anterior, representado por una medición de latencia muy inferior con el nuevo generador, de unos 13 segundos en promedio contra unos 54 segundos con el anterior. Esto da una razón de rendimiento de 4.4 o 337% de ventaja neta del nuevo generador sobre el anterior. Desde la perspectiva inversa, se puede decir que el nuevo generador tiene una latencia de solamente el 23% respecto al tiempo o latencia del generador anterior con lo cual se tiene una mejora del 77% al adoptar el nuevo generador de reportes.

Tabla 7: Resultados de Aplicación del Test 3.

TEST No	3							
TESTER	Generador de Reportes Anterior			Nuevo Generador de Reportes			n	n%
	Hora Inicio	Hora Fin	Tiempo	Hora Inicio	Hora Fin	Tiempo		
1	10:35:43	10:36:38	55	10:25:39	10:25:51	12	4.6	358%
2	9:39:12	9:40:06	54	9:42:26	9:42:39	13	4.2	315%
3	12:41:47	12:42:37	50	12:47:44	12:47:55	11	4.5	355%
4	9:00:07	9:01:00	53	9:07:10	9:07:22	12	4.4	342%
5	11:25:55	11:26:52	57	11:29:01	11:29:14	13	4.4	338%
6	3:05:27	3:06:19	52	3:07:34	3:07:46	12	4.3	333%
7	1:55:02	1:56:07	65	2:03:12	2:03:27	15	4.3	333%
8	11:20:32	11:21:28	56	11:24:13	11:24:25	12	4.7	367%
9	2:17:48	2:18:39	51	2:27:04	2:27:17	13	3.9	292%
10	2:17:08	2:18:00	52	2:22:00	2:22:12	12	4.3	333%
MEDIA			54	13			4.4	337%

La Gráfica 3 a continuación muestra la relación entre los resultados de la ejecución del Test 3 con el generador anterior y el nuevo generador de reportes.



**Gráfico 3: Resultados de la aplicación del Test 3.**

En conclusión, el nuevo generador de reportes tiene una consistente mejora de rendimiento sobre el anterior en todas las pruebas realizadas con mejora de la eficiencia entre el 67% al 77% sobre el anterior.

## **6.4.2 Evaluación de Facilidad de uso.**

### **6.4.2.1 Pasos de procedimiento**

La tabla a continuación muestra la cantidad de pasos de procedimiento necesarios para lograr la generación de cada prueba o test en cada generador de reporte. Como puede apreciarse, el nuevo generador requiere menos pasos para completar la tarea en todos los casos, concluyéndose que requiere en promedio 1.3 pasos menos que el anterior, por lo cual es más fácil de usar.

**Tabla 8: Pasos de Procedimiento.**

Caso de Prueba	Generador Anterior	Generador Nuevo
	Cantidad de Pasos	Cantidad de Pasos
Test 1	7	6
Test 2	9	7
Test 3	6	5
<b>Media</b>	<b>7.3</b>	<b>6.0</b>

#### 6.4.2.2 Simplicidad de uso

Se colectaron los datos vía ficha de registro de resultados de los 10 usuarios que ejecutaron la prueba, coincidiendo todos que ambas interfaces están bien diseñadas, son claras y fáciles de usar. Ambas son intuitivas, de complejidad simple para su comprensión y uso. El hecho que todos los testers eran usuarios expertos, es un factor que puede estar condicionando estos resultados, asimismo el hecho que ambas interfaces son muy similares, y en el caso de la del servidor de reportes anterior, la usan de manera cotidiana.

En vista que todos los resultados fueron coincidentes, se optó por no presentar en este informe la tabla de datos individuales, y solo se presenta la moda para cada resultado de evaluación (misma que es absoluta al coincidir la repetición de datos en un 100% con N<sup>16</sup>).

La tabla 9 muestra el resultado de la evaluación, en la que se evidencia que ambas interfaces resultan simples de usar para el operador.

Tabla 9: Resultado de Simplicidad de Uso.

Casos de Prueba	Generador Anterior	Generador Nuevo
	Simplicidad {Complejo, Simple}	Simplicidad {Complejo, Simple}
Test 1	Simple	Simple
Test 2	Simple	Simple
Test 3	Simple	Simple
<b>Moda</b>	<b>Simple</b>	<b>Simple</b>

En conclusión, el nuevo generador de reportes es un poco más fácil de usar que el anterior por cuanto tiene menos pasos procedimentales que realizarse para realizar la generación del reporte.

---

<sup>16</sup> En este contexto N corresponde al número de datos de la muestra.

## **7 Conclusiones y Recomendaciones**

Del análisis realizado a los resultados de todo el proceso, tanto de desarrollo del software como de evaluación de su uso, se desprenden las siguientes conclusiones y recomendaciones:

### **7.1 Conclusiones**

- Se logró obtener información y realizar un análisis satisfactorio de las necesidades y requerimientos de la Dirección General de Inversión Pública para la implementación de un Generador de Reportes para el Banco de Proyectos, gracias al involucramiento decidido del personal técnico a cargo del desarrollo del software del SNIP.
- Se logró definir y adoptar por parte el equipo, asimismo, las restricciones en cuanto a estándares de notación, codificación, herramientas y prácticas a observar en el desarrollo del proyecto, de acuerdo a las normativas de la Dirección de Desarrollo del Banco de Proyectos de la Dirección General de Inversión Pública, lo cual garantizó una integración expedita del nuevo Generador de Reportes al Banco de Proyectos del SNIP.
- La metodología RUP resultó ser fácilmente adaptable y útil para guiar la realización del presente proyecto, en todo su ciclo de vida, permitiendo una comprensión profunda y detallada de los requerimientos, necesidades y reglas del negocio a través de los modelos de casos de uso, dominio, clases y secuencia.
- Se elaboró un nuevo diseño para el generador de reportes que optimiza la forma en que son definidos y generados los reportes acorde a las normativas definidas por la Dirección General de Inversión Pública.

- Se hizo una implementación exitosa de la Herramienta Diseñadora de Reportes, siendo ésta el primer producto requerido para la operación del módulo generador de reportes, en la cual se realiza la definición y liberación de los reportes.
- La facilidad de uso de la Herramienta Diseñadora de Reportes, le permite eventualmente ser un instrumento para el usuario experto en el negocio, no necesariamente para un informático, como ocurre con la mayoría de los diseñadores de reportes disponibles.
- Se realizó con éxito el desarrollo de la clase que implementa la máquina generadora de reportes, cuya responsabilidad es construir el scripts que obtiene de la base de datos, la información requerida de acuerdo a los parámetros definidos por el usuario y la definición del reporte.
- Asimismo fue exitosa la implementación del Servidor de Reportes, logrando alcanzar los requerimientos de facilidad de uso para el usuario y una mejora en el desempeño de la generación de reportes y por ende en la productividad del usuario.
- En todas las pruebas realizadas se evidencia que el nuevo generador de reportes tiene una consistente mejora de rendimiento sobre el anterior con incremento de la eficiencia de 45% en cuanto al diseño de un nuevo reporte, y entre el 67% al 77% en cuanto a su uso para generar reportes.
- A partir de lo manifestado por los usuarios involucrados en la evaluación del nuevo generador de reportes se puede concluir que el nuevo generador de reportes es más fácil de usar que el anterior por cuanto requiere menos pasos procedimentales que realizarse tanto para diseñar nuevos reportes como para generarlos.

## **7.2 Recomendaciones**

- Se recomienda aprovechar el modelo de generación de reportes tabulares desarrollado para la implementación de generadores de reportes en los sistemas de información de otras instituciones con requerimientos similares a los abordados en este trabajo.
- Se recomienda a DGIP aprovechar las capacidades de la librería XlsIO para ampliar el alcance del tipo de reportes que se pueden generar con el módulo generador de reportes, permitiendo integrar otros objetos de Excel para producir reportes que incorporen múltiples secciones, gráficas, e inclusive tablas dinámicas.
- Se recomienda continuar la consolidación de la herramienta, a fin que madure lo suficiente para que pueda liberarse como proyecto de código abierto, u optimizarla con miras a una potencial comercialización.

Sobre la base de expuesto, se puede afirmar que se lograron los objetivos y beneficios esperados del nuevo módulo generador de reportes, y por tanto se puede confirmar que el trabajo de diploma Propuesta para la Implementación de Generador de reportes parametrizados para el Banco de Proyectos del Sistema Nacional de Inversión Pública se ha concluido con éxito, cumpliendo con las necesidades de la institución, mejorando las diferentes herramientas para el manejo de los reportes, haciéndolas más amigables para el usuario final y finalmente logrando una actualización del software que proporcionará una mejora significativa en el desempeño de la institución.

## 8 Bibliografía

- ABC, D. (s.f.). *Definición ABC*. Obtenido de Definición ABC:  
<http://www.definicionabc.com/tecnologia/informacion.php>
- Basu, C. (2013). What Are the Basic Kinds of Reports Produced by a Management Information System? *Chron - Small Bussiness (Demand Media)*, 12-14.
- Bootstrap. (1 de 1 de 2016). *Bootstrap main page*. Obtenido de Bootstrap web Site:  
<http://getbootstrap.com/>
- Corrales, C. F. (3 de 3 de 2014). *Desarrollo Tecnológico*. Obtenido de Prezi :  
<https://prezi.com/raax7o3ge0iq/desarrollo-tecnologico/>
- Escofet, C. M. (s.f.). *El lenguaje SQL*. uoc.edu.
- Gartner Corporation. (2015). *Magic Quadrant for Operational Database Management Systems* . New York, USA.: Gartner, Inc.
- Gerencie. (27 de Febrero de 2012). *Gerencie.com*. Obtenido de Gerencie.com:  
<http://www.gerencie.com/sistemas-de-informacion.html>
- Jane, L. V. (2007). Diccionario de la Lengua Española. En *Diccionario de la Lengua Española Manual Basico*. España: VERON | editores.
- Kenneth C. Laudon, Jane P. Laudon. (2008). *Sistemas de Informacion Gerencial:Administracion de la Empresa Digital*. En K. Laudon, & J. Laudon, *Sistemas de Informacion Gerencial* (pág. 14). Mexico: PEARSON EDUCACION.
- Lambert, Y. (31 de 05 de 2011). *Historia de Microsoft SQL Server – RDBMS*. Obtenido de Blog de Yamil Lambert Sarango: <http://blog.espol.edu.ec/ylambert/2011/05/31/historia-de-microsoft-sql-server-%E2%80%93-base-de-datos-relacional-rdbms/>
- Larman, C. (2003). *UML y Patrones: Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Madrid, España: Pearson - Prentice Hall.
- MacroH. (25 de 9 de 2012). *Características de Microsoft SQL Server 2008*. Obtenido de Tareas Universitarias: <http://itsoutside.blogspot.com/2012/09/caracteristicas-microsoft-sql-server.html>
- Martínez García, L. (8 de 2 de 2011). *Tipos de diseños de investigación*. Obtenido de Portal Web Centro Cochrane Iberoamérica:  
[http://www.cochrane.es/files/TipoDisenInvestigacion\\_0.pdf](http://www.cochrane.es/files/TipoDisenInvestigacion_0.pdf)

- Microsoft, inc. (1 de 1 de 2016). *SQL Server*. Obtenido de Portal corporativo Microsoft:  
<https://www.microsoft.com/es-es/server-cloud/products/sql-server/>
- Microsoft, Inc. (1 de 1 de 2016). *Visual Studio*. Obtenido de Portal de Visual Studio:  
<https://www.visualstudio.com/>
- MSDN Entity Framework. (5 de 12 de 2015). *Entity Framework Web Page*. Obtenido de MSDN - Data Access and Storage: <https://msdn.microsoft.com/es-ni/data/ef.aspx>
- Pressman, R. S. (2002). *Ingeniería de software un enfoque práctico*. España: Concepción Fernández Madrid.
- Ramos, I. A. (2009). *Herramienta Multimedia de apoyo a la Enseñanza de la Metodología RUP de Ingeniería del Software*.
- Syncfusion, Inc. (1 de 1 de 2016). *Syncfusion Essential Studio 2015*. Obtenido de Portal Web de Syncfusion: <http://www.syncfusion.com/products/whatsnew>
- Universidad de Charlotte. (22 de 05 de 2007). *INFO 2130. Introduction to Business Computing*. Recuperado el 28 de 10 de 2014, de TYPES OF INFORMATION SYSTEMS:  
<http://bisom.uncc.edu/courses/info2130/Topics/istypes.htm>
- Valdés, D. P. (26 de Octubre de 2007). *MAESTROS DEL WEB*. Obtenido de MAESTROS DEL WEB:  
<http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>



## 9 Anexos

### 9.1 ANEXO1. Formato o ficha de registro de resultados.

#### 9.1.1 Ficha de registro de resultados Creación de Reportes.

Registro de Resultados de Pruebas		
<b>Indicaciones:</b> Ejecute cada caso de prueba o Test, creando un nuevo reporte con la estructura definida en el formato del reporte actual indicado en cada caso. Cree el reporte primero con la herramienta de Diseño de reportes actual del Banco de Proyectos y luego con la nueva herramienta. En cada prueba anote los datos que se solicitan: <ul style="list-style-type: none"> <li>• <b>Tiempo:</b> hora de inicio y fin de la prueba (con minutos y segundos)</li> <li>• <b>Pasos:</b> Conteo de la cantidad de pasos que requieren ejecutarse en la herramienta para crear el reporte.</li> <li>• <b>Simplicidad:</b> Percepción suya de si el uso de la herramienta es simple o complejo. Considere si la interfaz de usuario está bien diseñada, es clara, fácil de entender, intuitiva, y los controles usados son conocidos y de fácil uso.</li> </ul>		
<b>Test1: Reporte financiero de Programación anual por tipo de recurso (No 03)</b>		
<b>Medida:</b>	<b>Generador Anterior</b>	<b>Generador Nuevo</b>
<b>Tiempo:</b>		
<b>Hora de Inicio:</b>		
<b>Hora finalización:</b>		
<b>Pasos de procedimiento:</b>		
<b>Simplicidad de uso:</b>	[ ] Simple, [ ] Complejo	[ ] Simple, [ ] Complejo
<b>Test2: Reporte Comparativo financiero de Programación anual vs ejecución (No 221)</b>		
<b>Medida:</b>	<b>Generador Anterior</b>	<b>Generador Nuevo</b>
<b>Tiempo:</b>		
<b>Hora de Inicio:</b>		
<b>Hora finalización:</b>		
<b>Pasos de procedimiento:</b>		
<b>Simplicidad de uso:</b>	[ ] Simple, [ ] Complejo	[ ] Simple, [ ] Complejo
<b>Test3: Reporte de Ejecución de Indicadores Físicos (No 333)</b>		
<b>Medida:</b>	<b>Generador Anterior</b>	<b>Generador Nuevo</b>
<b>Tiempo:</b>		
<b>Hora de Inicio:</b>		
<b>Hora finalización:</b>		
<b>Pasos de procedimiento:</b>		
<b>Simplicidad de uso:</b>	[ ] Simple, [ ] Complejo	[ ] Simple, [ ] Complejo

### 9.1.2 Ficha de registro de resultados Generación de Reportes.

Registro de Resultados de Pruebas		
<b>Indicaciones:</b>  Ejecute cada caso de prueba o Test, generando el reporte indicado primero con la herramienta de Diseño de reportes actual del Banco de Proyectos y luego con la nueva herramienta. Use los niveles predeterminados.  En cada prueba anote los datos que se solicitan: <ul style="list-style-type: none"> <li>• <b>Tiempo:</b> hora de inicio y fin de la prueba (con minutos y segundos)</li> <li>• <b>Pasos:</b> Conteo de la cantidad de pasos que requieren para generar el reporte.</li> <li>• <b>Simplicidad:</b> Percepción que tenga de la facilidad de uso del servidor de reportes, es simple o complejo. Considere si la interfaz de usuario está bien diseñada, es clara, fácil de entender, intuitiva, y los controles usados son conocidos y de fácil uso.</li> </ul>		
<b>Test1: Reporte financiero de Programación anual por tipo de recurso (No 03)</b>		
<b>Medida:</b>	<b>Generador Anterior</b>	<b>Generador Nuevo</b>
<b>Tiempo:</b>		
<b>Hora de Inicio:</b>		
<b>Hora finalización:</b>		
<b>Pasos de procedimiento:</b>		
<b>Simplicidad de uso:</b>	[ ] Simple, [ ] Complejo	[ ] Simple, [ ] Complejo
<b>Test2: Reporte Comparativo financiero de Programación anual vs ejecución (No 221)</b>		
<b>Medida:</b>	<b>Generador Anterior</b>	<b>Generador Nuevo</b>
<b>Tiempo:</b>		
<b>Hora de Inicio:</b>		
<b>Hora finalización:</b>		
<b>Pasos de procedimiento:</b>		
<b>Simplicidad de uso:</b>	[ ] Simple, [ ] Complejo	[ ] Simple, [ ] Complejo
<b>Test3: Reporte de Ejecución de Indicadores Físicos (No. 03)</b>		
<b>Medida:</b>	<b>Generador Anterior</b>	<b>Generador Nuevo</b>
<b>Tiempo:</b>		
<b>Hora de Inicio:</b>		
<b>Hora finalización:</b>		
<b>Pasos de procedimiento:</b>		
<b>Simplicidad de uso:</b>	[ ] Simple, [ ] Complejo	[ ] Simple, [ ] Complejo

## 9.2 ANEXO2. Guía de Referencia Técnica del Generador de Reportes.

### 9.2.1 Aspectos Generales

La máquina generadora de reportes es la que crea el script SQL que genera el reporte, esta se encuentra en la lógica del uso del SQL donde describiremos cada una de las funciones para llegar a la generación de los reportes tabulares.

Todas las tabla toman un papel importante en el reporte y se debe de tener en cuenta que todos los reportes tiene como origen de datos una función SQL de tipo tabla porque hay funciones que son de tipo escalar.

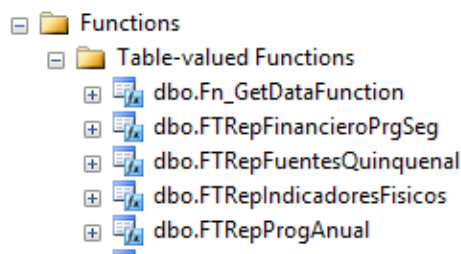


Ilustración 28, tablas tabulares

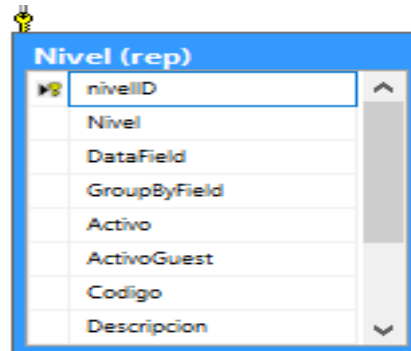
Los reportes están clasificados por categorías, estas categorías se encuentran alojadas en la tabla CategoriaReporte(rep), esta tabla solo posee el nombre de la categoría y el identificador único de la tabla. Entre las categorías de esta tabla podemos mencionar algunas por ejemplos: “Fuentes de financiamiento”, “Indicadores”, “Financiero-renglones”, entre otros.

CategoriaReporte (rep)	
	categoriaReporteID
	Nombre
	CodeInt
	Activo

Ilustración 29, tabla categoría

Los reportes tienen niveles de agrupación que están definidos por la función que está asociado al reporte que es de tipo tabla.

Los reportes poseen niveles, estos están definidos en la tabla Niveles (Rep.) es una tabla que contiene los catálogos de los niveles y que contienen otras columnas auxiliares que generan el script del reporte. Dentro de esta tabla encontramos diferentes campos que se ocupan para la generación del reporte.

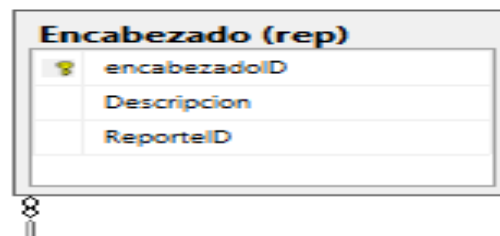


nivelID	Nivel	DataField	GroupByField	Activo	ActivoGuest	Codigo	Descripción

Ilustración 30, tabla niveles

Siendo esto podemos mencionar las tablas que toman un papel fundamental en la generación del reporte las cuales son:

La tabla encabezado es la tabla que funciona llenando los datos del encabezado del reporte el cual cada encabezado posee su id que lo enlaza con la tabla reporte (rep).



encabezadoID	Descripción	ReporteID

Ilustración 31, tabla encabezado

Una de las tablas que posee gran importancia y relevancia en la generación del reporte es la tabla Reporte(rep) ya que es la encargada de guardar el nombre del reporte que es la que se desplagara en la interfaz web, esta tabla no posee relación con el nombre del reporte que aparece en excel.

En esta tabla se encuentran ciertas columnas de configuración que funcionan de la siguiente manera el campo nombre guarda el nombre del reporte que se

mostrara en la interfaz web, el campo publicado acepta valores booleanos que permiten saber si el reporte se encuentra publicado, el campo categoriareporteid permite saber a que categoria pertenece el reporte y finalmente el campo parametrizado indica si los parametros dentro de la funcion estan enlazados con el control en la web.



Ilustración 32, Tabla reporte

La tabla NiveReporte(rep) se guardan todos los niveles que trae la función de tipo tabla la que está asociado a la tabla reporte(rep). Finalmente nivelreporte(rep) guarda los niveles que trae las funciones.

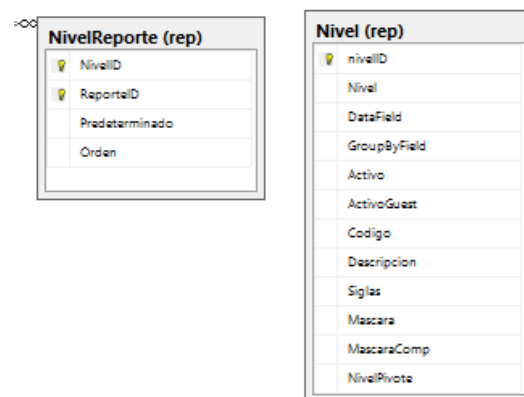
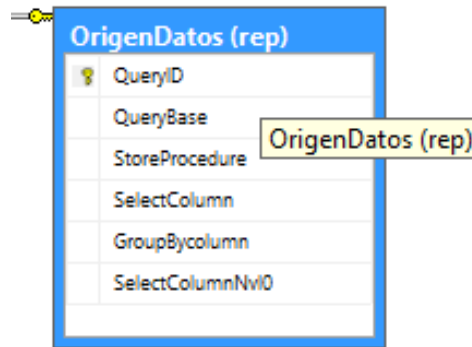


Ilustración 33, tablas de niveles

La tabla Origendedatos(rep) en esta tabla se guarda el nombre de la función de tipo tabla que pertenece el reporte y se guardan el procedimiento almacenado que

ejecuta la función además de las columnas de la función que son seleccionadas en el grid que se encuentra en el diseñador de reportes y las columnas que son sumadas en el diseñador, esto se abordara más a fondo durante el desarrollo del siguiente acápite.



The diagram shows a table titled "OrigenDatos (rep)" with a primary key icon on the left. The table has six columns: QueryID, QueryBase, StoreProcedure, SelectColumn, GroupBycolumn, and SelectColumnNvl0. A callout box labeled "OrigenDatos (rep)" points to the table.

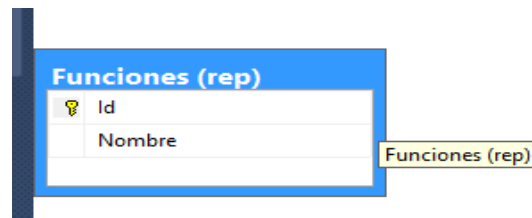
QueryID	QueryBase	StoreProcedure	SelectColumn	GroupBycolumn	SelectColumnNvl0
---------	-----------	----------------	--------------	---------------	------------------

Ilustración 34, Tabla origen de datos

### 9.2.2 Configuración

Esta sección corresponde a aquellas tablas que se guarda la configuración de los parámetros de la función con los controles en la interfaz web. Es decir que control web corresponde a que parámetro de la función de tipo tabla.

Existe la tabla funciones la cual su principal rol es almacenar el nombre de las funciones que se crean cuando se crean los reportes desde la herramienta diseñadora de reporte.

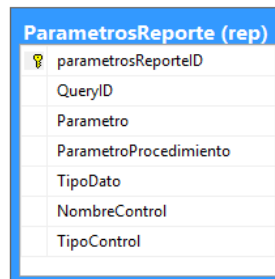


The diagram shows a table titled "Funciones (rep)" with a primary key icon on the left. The table has two columns: Id and Nombre. A callout box labeled "Funciones (rep)" points to the table.

Id	Nombre
----	--------

Ilustración 35, tabla funciones

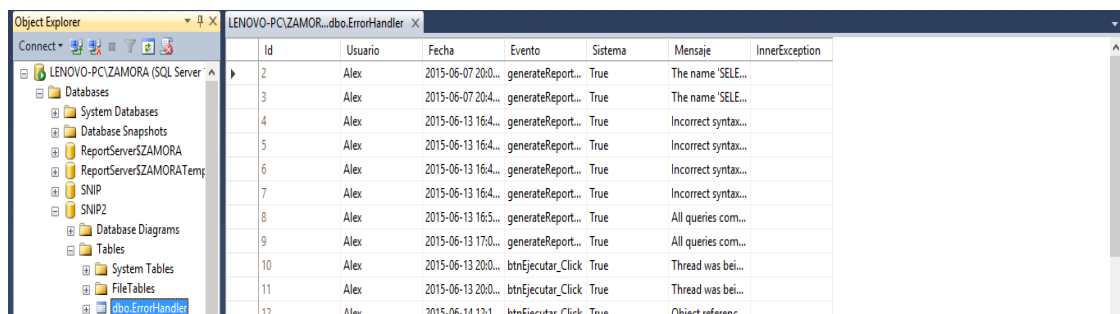
La segunda tabla que integra este conjunto es la tabla Parametrosreporte(rep) que es la encargada de almacenar los parámetros de la función y el tipo de datos que posee la función y a que control de interfaz web pertenece.



parametrosReporteID	QueryID	Parametro	ParametroProcedimiento	TipoDato	NombreControl	TipoControl
---------------------	---------	-----------	------------------------	----------	---------------	-------------

**Ilustración 36, tabla parámetros del reporte**

La tabla error handler es la encargada de manejar los errores producidos durante la generación del reporte o cualquier mal función que evite el correcto funcionamiento de la aplicación, los errores se pueden consultar desde la tabla y verificar el tipo de error que se está produciendo.



Id	Usuario	Fecha	Evento	Sistema	Mensaje	InnerException
2	Alex	2015-06-07 20:0...	generateReport...	True	The name 'SELE...	
3	Alex	2015-06-07 20:4...	generateReport...	True	The name 'SELE...	
4	Alex	2015-06-13 16:4...	generateReport...	True	Incorrect syntax...	
5	Alex	2015-06-13 16:4...	generateReport...	True	Incorrect syntax...	
6	Alex	2015-06-13 16:4...	generateReport...	True	Incorrect syntax...	
7	Alex	2015-06-13 16:4...	generateReport...	True	Incorrect syntax...	
8	Alex	2015-06-13 16:5...	generateReport...	True	All queries com...	
9	Alex	2015-06-13 17:0...	generateReport...	True	All queries com...	
10	Alex	2015-06-13 20:0...	btnEjecutar_Click	True	Thread was bei...	
11	Alex	2015-06-13 20:0...	btnEjecutar_Click	True	Thread was bei...	
12	Alex	2015-06-14 12:1...	btnEjecutar_Click	True	Object referenc...	

**Ilustración 37, tabla manejadora de errores**

La query que se genera de manera dinámica está formada por una llave especial de agrupación que se crea de forma dinámica en base a los niveles de agrupación seleccionados por el usuario desde la interfaz web, esta llave sirve para ordenar el resultado y muestra la información en el orden deseado las columnas seleccionadas que se mostraran en el reporte las columnas por las cuales se van a agrupar en base a los niveles de agrupación seleccionados.

### 9.2.3 Funciones

Dentro de las funciones que componen la maquina generadora de reportes se encuentran los siguientes valores de funciones escalares describiendo los siguientes incisos:

- ✓ **generate\_key:** esta función escalar genera la llave compuesta en base al nivel de agrupación que se está trabajando en ese momento y que sirve para ordenar la información.
- ✓ **generate\_select:** esta función trae las columnas seleccionadas, es decir las columnas que se van a mostrar en base al reporte.
- ✓ **generate\_groupby:** esta función trae las columnas por las cuales se agrupara el reporte, es decir las columnas que corresponden a los niveles de agrupación seleccionados en base al reporte.
- ✓ **generate\_query:** es un procedimiento almacenado que en conjunto con las funciones anteriormente mencionadas, se encarga de armar la query final con la cual se obtendrá la información requerida para el reporte.
- ✓ **Showcolumns:** es una función auxiliar, que se creó para los reportes de tipo indicadores, básicamente indica si se debe de incluir o no ciertas columnas desde la aplicación web se envía al SQL los parámetros que necesitan el reporte, los cuales son los mismos que necesita la función, además del nombre de la función.

### 9.2.4 Procedimientos Almacenados

**PA GenerateQuery:** Este procedimiento almacenado recibe los siguientes parámetros @niveles varchar(25), @reportID int, @from varchar(255).

@niveles: Son los niveles por los cuales se va a agrupar el reporte cabe destacar que esto es de forma dinámica de acuerdo a las necesidades que tenga el usuario



para la generación del reporte, recibe una cadena de caracteres los cuales son los niveles separados por coma (,).

@reportID: Es el identificador (ID) de la tabla de los reportes que corresponde al reporte seleccionado por el usuario final, recibe tipo de datos enteros.

@from: corresponde a la cláusula “from” de la función de tipo tabla que corresponde al reporte seleccionado y que lleva los parámetros que este recibe.

El procedimiento almacenado hace uso de las funciones explicadas anteriormente en la sección 5.1.2 y de los parámetros que recibe el procedimiento para construir de manera dinámica y en el orden requerido, agrupado por los niveles proporcionados la consulta SQL que ejecutara y retornara el conjunto de datos los cuales serán enviados a la aplicación web para su procesamiento y la construcción del reporte Excel que se generara para el usuario final.

**PA GetFunctionList** Este procedimiento almacenado recibe los siguientes parámetros @categoryId varchar, @todas int.

@ categoryId: Este parámetro es el identificador de la categoría que recibe una cadena de caracteres.

@todas: Corresponde a un campo bandera para mostrar todas las funciones que pertenecen a la categoría y si han sido parametrizados o no.

Este procedimiento almacenado tiene la función de obtener la lista de las funciones que ya han sido parametrizadas y las que aún no son filtradas por la categoría de acuerdo a la bandera que recibe.

**PA GetFunctionParameters** Este procedimiento almacenado es el encargado de obtener en base al nombre de la función la lista de parámetros que la función recibe separados por coma. Recibe el parámetro @Functionname con una cadena de caracteres.

**PA GetFunctionParametersById:** Este procedimiento almacenado retorna una tabla con las siguientes columnas, nombre de la función, parámetros de la función, tipo de dato, tipo de control, nombre del control estas dos últimas están relacionadas con la interfaz web. Recibe el parámetro @functionName varchar el cual es una cadena de caracteres.

**PA Lista Reportes:** Muestra la Lista de Reportes Activos que se han creado, este procedimiento no recibe ningún parámetro.

**PA Nuevas Funciones:** Muestra la lista de funciones que aún no han sido parametrizadas. Este procedimiento no recibe ningún parámetro.

**PA SaveLevelByReport:** Este procedimiento almacenado recibe los siguientes parámetros @reportID int, @levels varchar, @Defaultlevels varchar.

@reportid anteriormente se había hablado de este identificador que cumple la misma función.

@levels Es la lista de niveles que la función de tipo de tabla contiene.

@Defaultlevels Es la lista de niveles seleccionados por el usuario para cargarse de manera predeterminada en la interfaz web.

Este procedimiento almacenado guarda todos los niveles que la función de tipo table tiene además de guardar cuales niveles se cargaran de manera predeterminada elegidos por el usuario final al momento de crear, actualizar o duplicar el reporte.

### 9.3 **ANEXO 3. Manual de Usuario de la Herramienta Diseñadora de Reportes.**

En el desarrollo de las diferentes necesidades que solicita el banco de proyectos se desarrolló la herramienta diseñadora de reportes la cual ayuda a la gestión de los diferentes reportes que se generan para la institución para esto se tomó los principales puntos a mejorar y se decidió crear una herramienta de escritorio que ayudara en la necesidad de administración y creación de los nuevos reportes.

Esta solución está dividida en diferentes secciones según el tipo de información que se vaya a manejar.

Primeramente se presenta lo que es la interfaz inicial del programa el cual está dividido de la siguiente manera la cual se detalla a continuación:

En la parte superior del programa se muestra el nombre del programa junto al icono del SNIP para que se identifique con facilidad al tipo de institución que pertenece además del uso que se le da al programa.

Además presenta los 3 tipos de opciones que se puedes hacer dentro del programa como es la creación de reportes, manejo de los reportes atreves de la configuración de los reportes, y también permite la administración de los catálogos que se necesitan durante la generación del reporte.



Ilustración 38, GUI de la Herramienta generadora de reportes

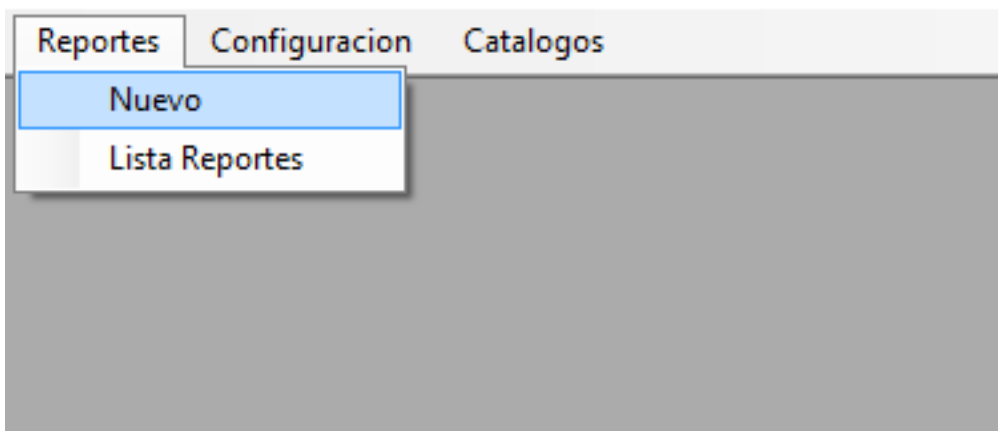


Ilustración 39, GUI reportes

Dentro del menú reportes se muestra dos de los tipos de opciones que se pueden escoger para el manejo de reportes.

Cuando se selecciona la opción nuevo se desplegará una ventana mostrando diferente tipos de opciones en las que se puede interactuar lo primero que se muestra son los reportes que se tienen en el momento.

1-Lista de reportes

2-botone nuevo, editar y duplicar

3-Cantidad de reportes dentro de la base de datos.

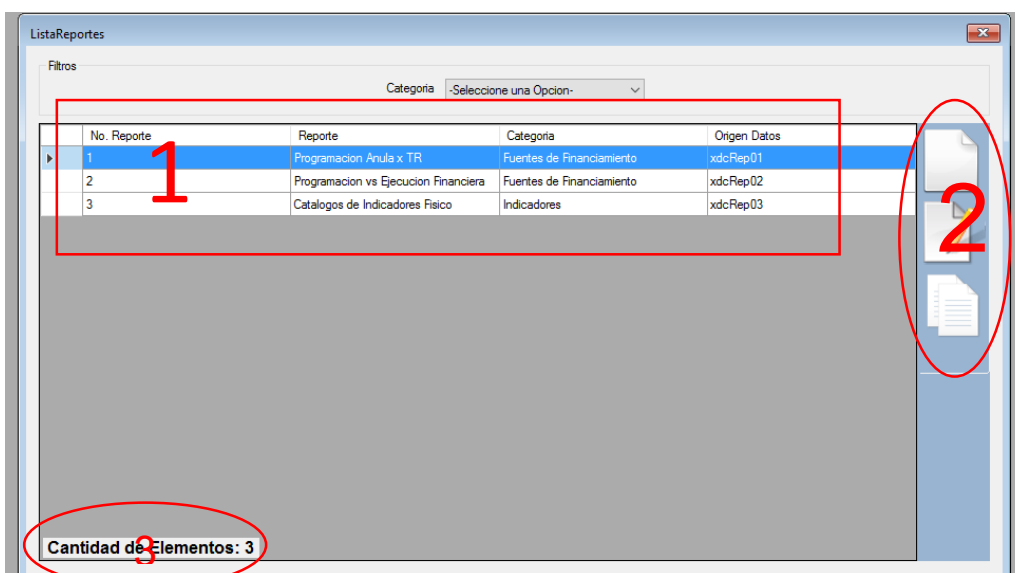


Ilustración 40, GUI Nuevo

Dentro de listas de reportes se muestran los diferentes reportes generados y también la manera de cómo se pueden administrar, esto se logra mediante los

botones que se encuentran en la parte derecha los cuales son: nuevo, editar y duplicar, también posee la opción de ver los reportes por categoría al cual cuando se selecciona la categoría solo los reportes pertenecientes a la categoría son ordenados.

En la siguiente imagen se muestra un ejemplo de cuando se selecciona la categoría indicadores y como la herramienta diseñadora de reportes ordena solo el reporte correspondiente a esta categoría, este reporte es mostrado sin problemas en el grid también se debe de tener en cuenta que en la parte inferior se muestra la cantidad o el número de reportes totales en la categoría seleccionada.

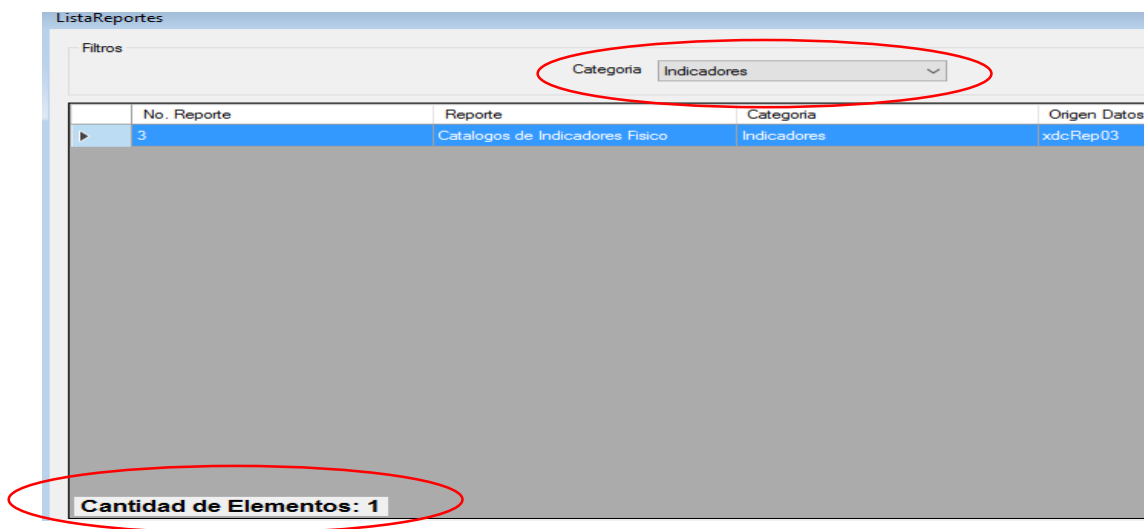


Ilustración 41, Lista de reportes

Seleccionado el reporte que se desea se pueden realizar una de las siguientes opciones como son: modificar el reporte con el icono “modificar” se tocara más adelante de esta sección como realizar una modificación exitosa de un reporte con más detalle y los pasos a seguir.

La siguiente acción que se puede realizar es la opción “duplicar” de igual forma este temas se tocara con más detalle ya que de la misma forma posee pasos específicos los cuales se deben de seguir.

A manera de aclaración para el usuario final se tiene que poner el reporte seleccionado en color azul ya que de lo contrario no se podrá ejecutar la acción ya que el programa necesita un objetivo para saber de qué reporte se está trabajando.

## Como editar un reporte

The screenshot shows the 'Edición de Reportes' window. It includes a toolbar with 'Guardar' and 'Salir' buttons. The 'Lista de Funciones' section has a dropdown for 'Orígenes de Datos Disponibles' set to 'FTRepIndicFisicos'. The 'Otros Datos' section contains fields for 'Nombre del Reporte' (Catalogos de Indicadores Fisico), 'Categoria' (Indicadores), and 'Encabezado del Reporte' (Indicadores Fisicos). The 'Niveles' section on the left lists various categories with checkboxes, including 'FTRepIndicFisicos', 'CodSectorInst', 'CodInstitucion', 'CodRealInst', 'CodSubsector', 'CodSector', 'CodProyecto', 'CodSubproyecto', 'CodActividad', 'CodOtra', and 'CodTipoProy'. The 'Niveles Predeterminados' table on the right lists levels 1 through 4 with corresponding codes. The 'Columnas' section on the left lists report columns with checkboxes, including 'FTRepIndicFisicos', 'Siglas', 'FechaRealInicio', 'FechaRealFin', 'Año', 'Costo', 'Cantidad', and 'Indicador'. The 'Columnas del Reporte' table on the right shows columns for 'Nombre', 'Sumatoria', and 'Indicador' with checkboxes for each.

Orden	Nivel
1	CodInstitucion
2	CodProyecto
3	CodSubproyecto
4	CodTipoProy

Nombre	Sumatoria
Indicador	<input type="checkbox"/>
Cantidad	<input type="checkbox"/>
Costo	<input checked="" type="checkbox"/>

Ilustración 42, GUI Edición de un reporte

Dentro de la edición del reporte se encuentran diferentes cuadros de dialogo donde se pueden seleccionar las opciones a modificar, primero se tiene la sección “orígenes de datos disponibles” esto indica el nombre de la función de tipo tabla con la que se está trabajando la cual se encuentre relacionada a un reporte, como se puede observar debajo de esta sección se encuentran los “niveles” que se deben de incluir siempre en cada reporte esta opción se puede seleccionar para que posterior aparezca en el primer grid de relleno de la derecha, de igual forma se muestran las “columnas” que componen valga la redundancia las columnas del reporte, las cuales pueden ser agregadas al grid simplemente con seleccionar el checkbox en el cuadro de dialogo “columnas” estas dentro del grid se puede seleccionar si se quiere sumarizar los valores.

En la parte superior del primer grid se encuentran 3 secciones que identifican al reporte “nombre del reporte” que define el nombre que poseerá el reporte, “categoría” es la sección que categoriza el reporte para después poder ser visualizado; El “encabezado del reporte” es el nombre que se desplegará cuando se habrá el documento Excel dentro del reporte ya generado. Finalmente se tienen los dos botones “guardar” que almacena cualquier cambio que se haya realizado en la edición del reporte y el botón “cerrar” que cierra el cuadro de edición de la aplicación.

## **Como realizar una duplicación de reporte**

The screenshot shows the 'Edición de Reportes' window with the following configuration:

- Lista de Funciones:** Orígenes de Datos Disponibles: FTRepProgAnual.
- Otros Datos:** Nombre del Reporte: Programacion Anual x TR; Categoría: Financiero; Encabezado del Reporte: Programacion Anual Por Tipo de Recurso.
- Niveles:** FTRepProgAnual.
- Niveles Predeterminados:** Orden 1: Institucion, 2: Proyecto, 3: Subproyecto, 4: Fuente.
- Columnas:** FTRepProgAnual.
- Columnas del Reporte:** A table with columns 'Nombre' and 'Sumatoria'. The rows are: Propios, Tesoro, Donacion, Prestamo, and MoAnual. All 'Sumatoria' checkboxes are checked.

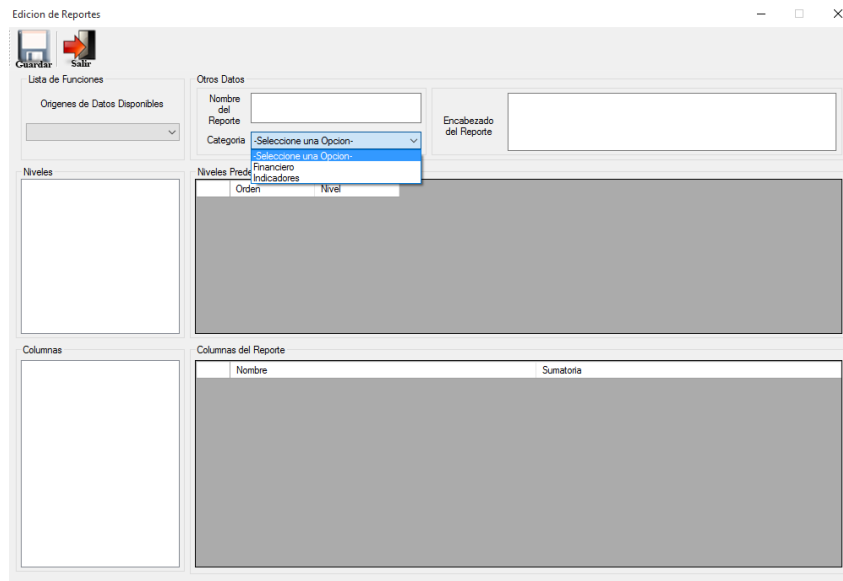
**Ilustración 43, Duplicidad de un reporte**

La opción de duplicar reporte posee la misma interfaz de cuadros de diálogos que la opción de edición diferencia que no muestra varios de los campos que anteriormente se encontraban con información.

Dentro de las opciones que se deben de tomar en cuenta para la generación satisfactoria del reporte son:

1. En origen de datos seleccionar el tipo de función de tipo tabla con la que se está trabajando para la duplicidad del reporte.
2. Seleccionar los niveles que serán agregados en el primer grid.
3. Las columnas que deberán de seleccionar deben ser iguales al reporte a duplicar para eso se tiene que tener previamente conocimiento de las columnas que usa el reporte a duplicar esta opción es útil debido si se desea tener más columnas a un reporte ya existente.
4. Seleccionar en el grid las columnas que tendrán suma
5. Colocar el nombre del reporte duplicado
6. Seleccionar la categoría a la que va a pertenecer el reporte
7. Seleccionar el encabezado del reporte el cual se mostrara cuando se genere el archivo Excel.
8. Guardar el reporte y verificar que se muestra en la lista de reportes.

## GUI manejo de reportes



**Ilustración 44, Nuevo reporte**

Cuando se selecciona la opción nuevo reporte se presenta la misma interfaz que se viene presentando en explicaciones pasadas, ya que se quiso hacerlo lo más sencillo para el usuario final y no estuviera que estar contemplando diferentes formas de modificar o realizar un reporte, de igual manera se creó un acceso más cómodo para el usuario desde el primer inicio de la aplicación para cuando decida realizar un nuevo reporte.



**Ilustración 45, Acceso fácil a nuevo reporte**

Como anteriormente se ha venido explicando para realizar un nuevo reporte se puede acceder de dos maneras la primera es desde la primera vez que se ejecuta la aplicación y la segunda cuando se dirige a lista de reportes desde el botón nuevo, sabiendo esto se pasara al siguiente punto donde se explicara con detalle la forma correcta de generación de un reporte.



### Como realizar un nuevo reporte

La realización de un reporte se hace de la siguiente manera primero se accede a la interfaz de nuevo reporte donde se selecciona el tipo de función tabular que dependerá el reporte los pasos a seguir son los siguientes:

1. Seleccionar la función tabular del reporte.
2. Nombrar al reporte, seleccionar a la categoría que estará asignado el reporte y nombrar el encabezado cuando se genere el reporte Excel.
3. Seleccionar los niveles que compondrán los niveles que se mostraran por defecto en la página web, esto es marcar los checkbox en la parte izquierda.
4. Seleccionar las columnas que estará formado el reporte, posterior dar check a las columnas que harán sumatoria en el reporte.
5. Guardar el reporte y esperar el mensaje de confirmación que el reporte se guardó con éxito.

The screenshot shows the 'Edición de Reportes' (Report Editing) window. It contains several sections with red boxes and labels indicating the steps for creating a new report:

- Paso 5:** A box around the 'Guardar' (Save) and 'Salir' (Exit) buttons at the top left.
- Paso 1:** A box around the 'Orígenes de Datos Disponibles' (Available Data Sources) dropdown menu.
- PASO 2:** A box around the 'Nombre del Reporte' (Report Name), 'Categoría' (Category), and 'Encabezado del Reporte' (Report Header) fields.
- PASO 3:** A box around the 'Niveles' (Levels) and 'Niveles Predeterminados' (Default Levels) sections.
- Paso 4:** A box around the 'Columnas' (Columns) and 'Columnas del Reporte' (Report Columns) sections.

Ilustración 46, Crear un nuevo reporte

A continuación el usuario se debe de dirigir a **configuración para la configuración de los parámetros** de los reportes con los respectivos formatos de cada parámetro.

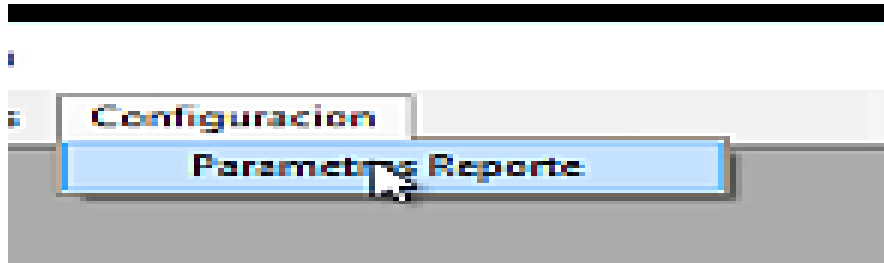


Ilustración 47, Ubicación parámetros de reporte

Cuando se despliegue parámetros del reporte lo que se debe de seleccionar es la función que se ha seleccionado anteriormente para la creación del reporte y de igual forma hay que darle Click a modificar para agregar los parámetros que necesita el reporte para la generación del mismo.

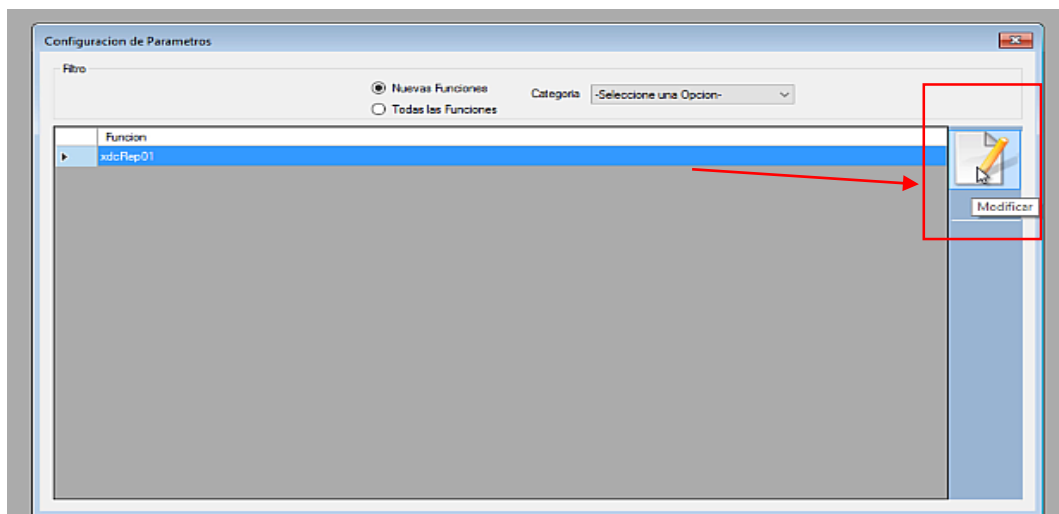


Ilustración 48, Modificar parámetros de un reporte

Cuando se da Click en el botón modificar aparecen diferentes tipos de datos que deben ser rellenados de la siguiente manera:

1. Se tiene que escribir el nombre de la función
2. Se debe escribir el nombre del parámetro
3. Se debe de seleccionar el tipo de dato que puede ser
4. Se selecciona el tipo de control
5. Se selecciona actualizar
6. Este mismo proceso se debe de realizar según el número de funciones que se tengan.

7. Finalmente se da guardar y se espera el mensaje de confirmación de que se guardaron los datos satisfactoriamente.
8. Se puede verificar el nuevo reporte desde lista de reportes

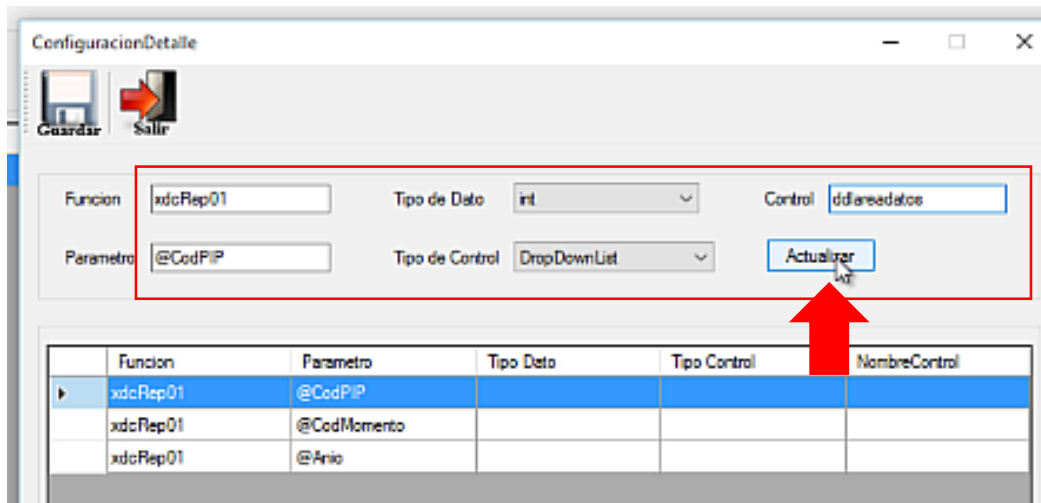


Ilustración 49, GUI edición parámetros del reporte

## Catálogos

Finalmente se tiene la opción catálogos la cual ayuda a la modificación y creación de las diferentes categorías que pueden tener los reportes y los diferentes niveles que pueden ser modificados de una manera sencilla y amigable.

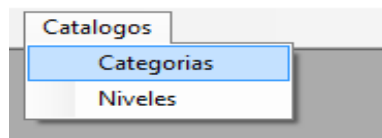


Ilustración 50, Ubicación de categorías en catálogos

### Actualizar un catalogo

Dentro de la opción categoría se pueden modificar los datos que componen a los reportes con la opción “actualizar” esto se logra siguiendo los siguientes pasos que son:

1. Seleccionar la categoría que se desea modificar
2. Cambiar el nombre si se desea hacer un cambio
3. Modificar el código que no es más que una abreviación del nombre de la categoría.
4. Tocar el botón actualizar para que se guarden los cambios realizados.

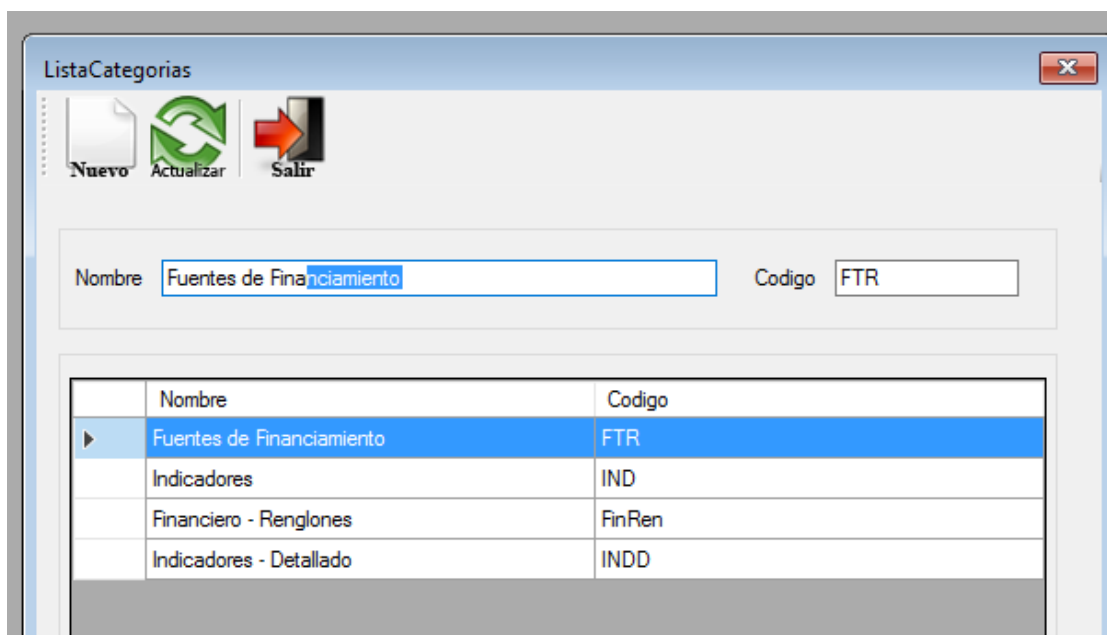


Ilustración 51, Actualización de categorías

### Creación de una nueva categoría

Si en cambio se desea realizar una nueva categoría se tiene que dar Click primeramente al botón “nuevo” posterior se debe de rellenar los campos que componen la categoría como son: nombre de categoría y código, al completar esta acción se tiene que dar al botón “guardar” donde se mostrara la nueva categoría en el grid de la aplicación a manera de ejemplo se creó una categoría con el nombre test.

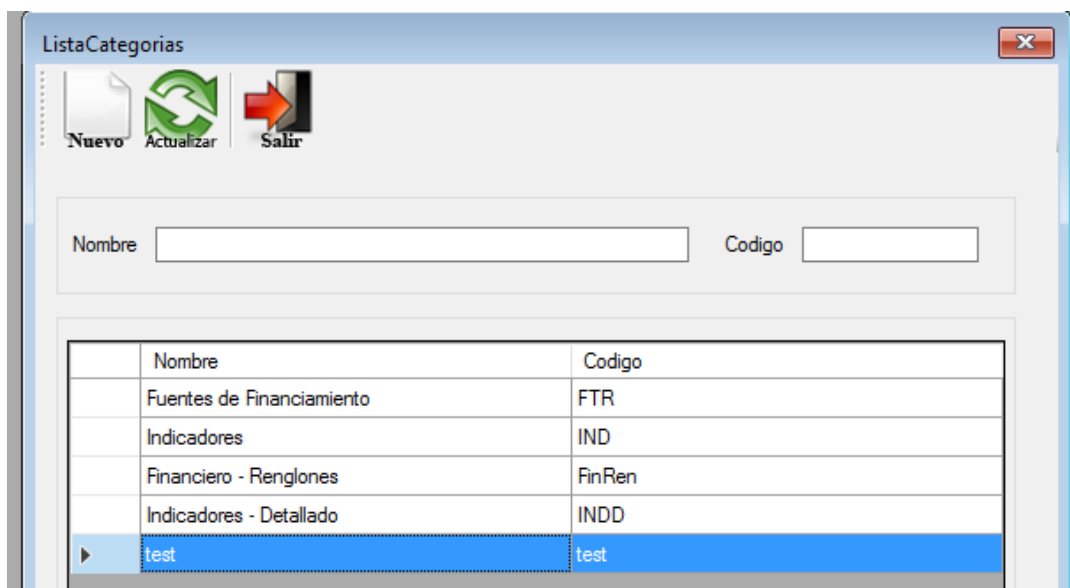


Ilustración 52, Creación de nueva categoría

Finalmente se tiene lo que es la edición de los niveles en la opción de catálogos los niveles están compuestos por diferentes campos de textos los cuales tiene la función de mostrar la edición de un nivel o así mismo la creación de un nivel.

### Edición de niveles

Para la edición de un nivel simplemente se tiene que hacer selección dentro del nivel a editar de esta manera se rellenaran los diferentes campos de texto lo cual permite editar la opciones que son: nivel, mascara, Mascara Comp, código, descripción, group field, también permite checkear lo que es activo, activo guest, nivel pivote. De esta manera se puede realizar la modificación necesaria y finalizar al dar en el botón “actualizar” donde se mostrara un mensaje que el nivel se ha actualizado con éxito.

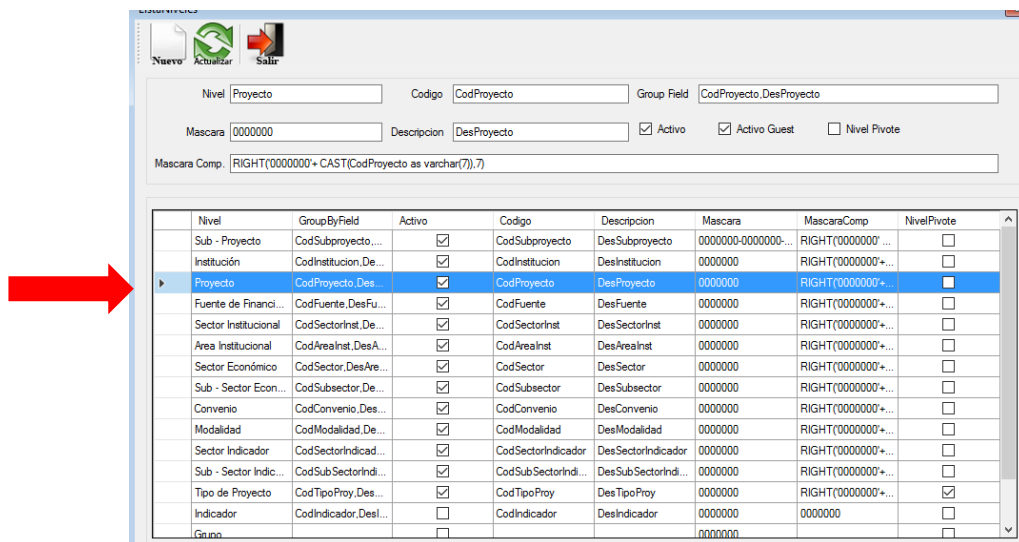


Ilustración 53, Edición de niveles

Para la creación de un nuevo nivel se tiene que dar primeramente en el botón nuevo y rellenar los campos que se tocaron anteriormente como son nivel, mascara, Mascara Comp, código, descripción, group field, también se debe checkear lo que es activo, activo guest, nivel pivote para finalmente darle al botón guardar el cual mostrara un mensaje con la leyenda nivel guardado.

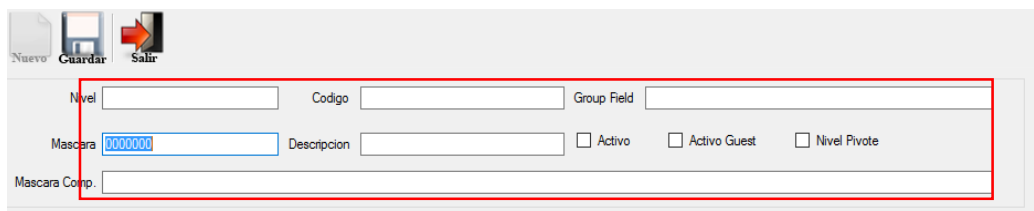


Ilustración 54, Creación de nuevo nivel

## 9.4 ANEXO 4. Manual de Usuario del Servidor de Reportes

La interfaz gráfica es uno de los aspectos más importantes en el desarrollo de este proyecto ya que es el producto donde el usuario final podrá generar los reportes así mismo la forma en que se presentaran y la forma en que estarán organizados. La interfaz web presenta la siguiente disposición en la parte superior se encuentra el banner de la institución, en la parte lateral izquierda un menú vertical con opciones de navegación y en el centro llamada banco de proyectos la finalidad de la página que es la generación de reportes.

Como se puede observar en la parte superior de la página la interfaz gráfica cuenta con un banner de la institución pública del SNIP el cual contiene el logo del SNIP una leyenda que describe la función de la institución y símbolos patrios que realzan los colores de la página, este modelo web se viene trabajando en las últimas fases de actualización de la página.



Ilustración 55, Logo SNIP

De igual forma se encuentran dos enlaces los cuales uno llamado inicio redirecciona a la página principal de la institución y el botón invitado (que solicita las credenciales para un acceso de mayor nivel si el usuario trabaja o posee un rol en la institución.)

La siguiente es la sección lateral izquierda llamada opciones de consulta que contiene diferentes enlaces de navegación que direccionan a diferentes puntos de la página; En esta sección se puede seleccionar el botón Regresar para cargar la página anterior que se tenía antes de llegar a la sección de generación de reportes.

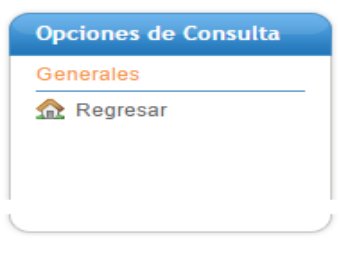


Ilustración 56, Opciones de consulta

La sección Banco de proyectos es la parte fundamental de la página ya que aquí se encuentran los principales controles que permiten seleccionar un reporte y generarlo en base a los filtros seleccionados que requiera.

La disposición de los controles es la siguiente: 4 subdivisiones, Tipo de reporte, reporte, niveles y filtros.

Cuando el usuario se encuentre en esta sección debe de recordar la herramienta diseñadora de reportes ya que está estrechamente relacionado de la forma en que se muestran los datos en esta sección.

### **La opción tipo de reporte**

Esta lista desplegable que muestra las categorías por las cuales están agrupadas los reportes y en base a la selección usada la lista de reportes va a variar ya que los reportes está agrupados por categorías.

The screenshot shows a web interface titled 'Banco de Proyectos'. Below the title is a section labeled 'Categoria de Reporte'. Inside this section is a dropdown menu currently showing 'Fuentes de Financiamiento'. Below the main interface, a separate dropdown menu is shown in an expanded state, displaying the following options: 'Fuentes de Financiamiento' (highlighted in blue), 'Indicadores', 'Financiero - Renglones', and 'Indicadores - Detallado'. At the bottom of this expanded menu is a placeholder option: '-Seleccione una Opcion-'.

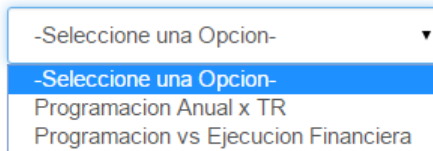
Ilustración 57, Categoría del reporte

### **La opción Reportes**

Esta lista desplegable es llenada en base al criterio de selección que se dio anteriormente y muestra los reportes que corresponden a la categoría seleccionada. Estos reportes son los construidos en la maquina diseñadora de reportes por el usuario final y son mostrados en la lista de reportes.



## Reporte



-Seleccione una Opcion-  
-Seleccione una Opcion-  
Programacion Anual x TR  
Programacion vs Ejecucion Financiera

Ilustración 58, opción reportes

### Niveles del reporte

Cada reporte posee niveles de agrupamiento y desde la maquina diseñadora de reportes se pueden definir niveles de agrupamiento predeterminado los que nos lleva a la siguiente sección.

En la imagen se puede observar dos columnas una llamada “niveles disponibles” y “niveles seleccionados”.

Los niveles disponibles son todos aquellos niveles por los cuales se puede agrupar un reporte seleccionado.

### Niveles



**Disponibles**

- Sector Institucional
- Area Institucional
- Sector Económico
- Sub - Sector Econó

**Seleccionados**

- Institución
- Proyecto
- Sub - Proyecto
- Fuente de Financia

Agregar  
Quitar  
Subir  
Bajar

Ilustración 59, niveles de reportes

Los niveles seleccionados son los niveles previamente seleccionados como predeterminados al momento de crear al reporte

Niveles Predeterminados		
	Orden	Nivel
▶	1	Institucion
	2	Proyecto
	3	Subproyecto
	4	Fuente

Ilustración 60, niveles de reportes SQL

El usuario cuando se encuentra en esta sección está en la capacidad de agregar más niveles o quitar niveles gracias a los botones que se encuentran en medio de esta columnas que permite mover el nivel seleccionado a las diferentes columnas (“seleccionados” a “disponibles” o viceversa) según desee la generación del reporte, además puede influir en el orden en que son mostrados los niveles con los botones de subir o bajar esto determina el orden cuando se genere el reporte permitiendo así más opciones para la generación dinámica del reporte.

### Parámetros

**Filtros**

---

Area de Datos: DGIP/Oficial ▼

Año:

**Ilustración 61, Parámetros de filtros de Datos y Tiempo**

La sección de los filtros se muestra de manera dinámica en base al reporte seleccionado previamente, desde la maquina diseñadora de reportes se tuvo que parametrizar los filtros del reporte esto quiere decir que se le indica cuales son los controles que desde la interfaz web se van a mostrar los cuales son los que el reporte necesita para generarse. En la figura se muestra que el reporte seleccionado necesita de estos dos parámetros.

ConfiguracionDetalle

Guardar
Salir

Funcion: FTRepProgAnual
 Tipo de Dato: int
 Control: DDLAreaDatos

Parametro: @CodPIP
 Tipo de Control: DropDownList
Actualizar

	Funcion	Parametro	Tipo Dato	Tipo Control	NombreControl
▶	FTRepProgAnual	@CodPIP	int	DropDownList	DDLAreaDatos
	FTRepProgAnual	@CodMomento	int	DropDownList	DDLAreaDatos
	FTRepProgAnual	@Anio	int	DropDownList	btAnio

**Ilustración 62, Parametros SQL**

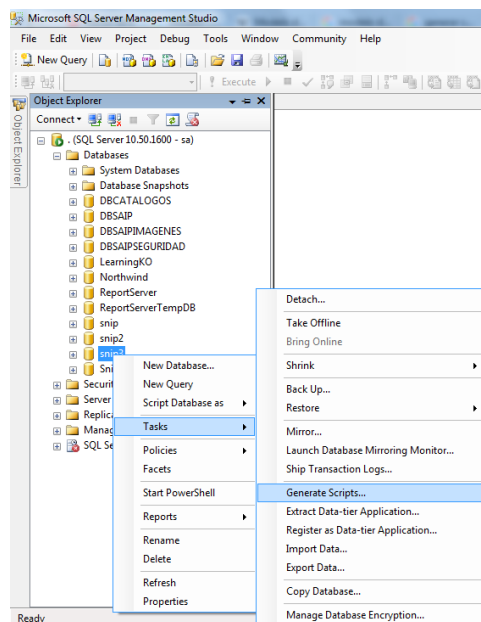
## 9.5 ANEXO 5. Procedimiento para exportación de la Base de Datos.

Una base de datos creada por una versión más reciente de SQL Server no puede ser restaurada a una versión anterior. La restricción se da simplemente porque una versión anterior desconoce sobre los cambios de formato que se introdujeron en la nueva versión.

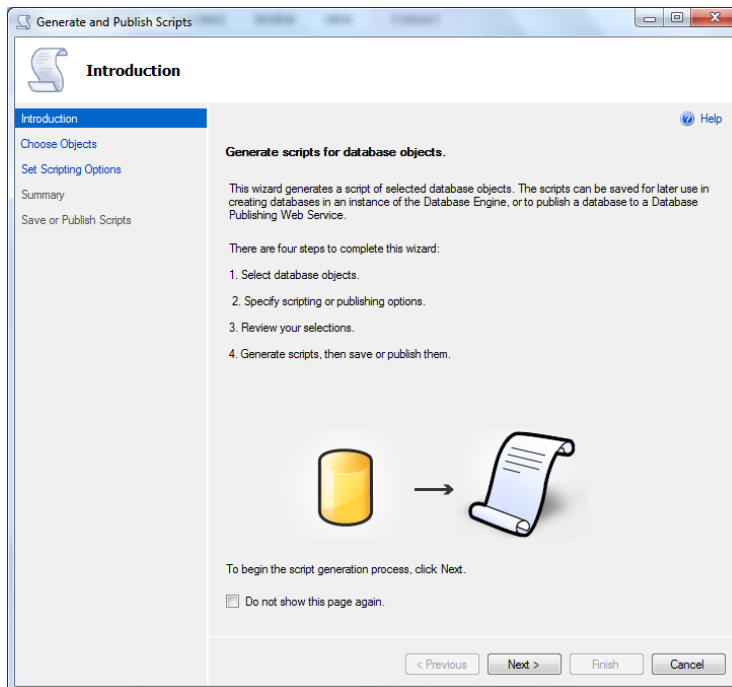
SQL Server mantiene unos números de versión interna que para el caso de SQL Server 2008 R2 es 661 y para el caso de SQL Server 2005 es 611.

Una solución es usar el asistente de generación de scripts de SQL Server 2008 R2. Para usarlo seguimos estos pasos:

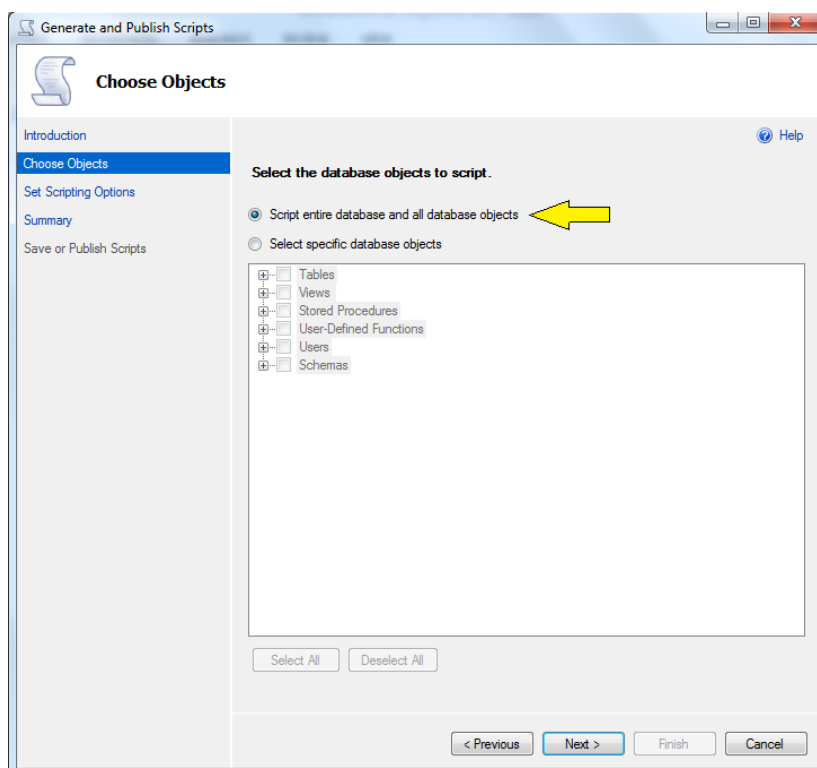
1. Ubicarse en la base de Datos que se requiere pasar a SQL Server 2005, dar clic derecho y seleccionar “Tasks” y luego “Generate Scripts” (Captura1).
2. Presenta la ventana (Captura2), clic en Next.
3. Aparece la siguiente ventana (Captura3), validar que este seleccionada la primera opción “Script the entire database and all database objects” y presionar Next.
4. Aparece la siguiente ventana (Captura4), Revisar la ruta del archivo a guardar, presionar el botón “Advanced”.
5. Aparece la siguiente ventana (Captura5), Seleccionar de la versión de SQL Server a la cual se creara el script desde la opción “Script for Server Version” y que el script contenga los datos y el esquema desde la opción “Type of Data to Script”; Presionar el boton “Ok” y clic en el botón Next.
6. Nos muestra la siguiente ventana (Captura6) que es un resumen de nuestra configuración y damos clic en Next, para que inicie a generar el script.
7. Cuando termina nos presenta la siguiente ventana (Captura7) y damos clic en Finish.



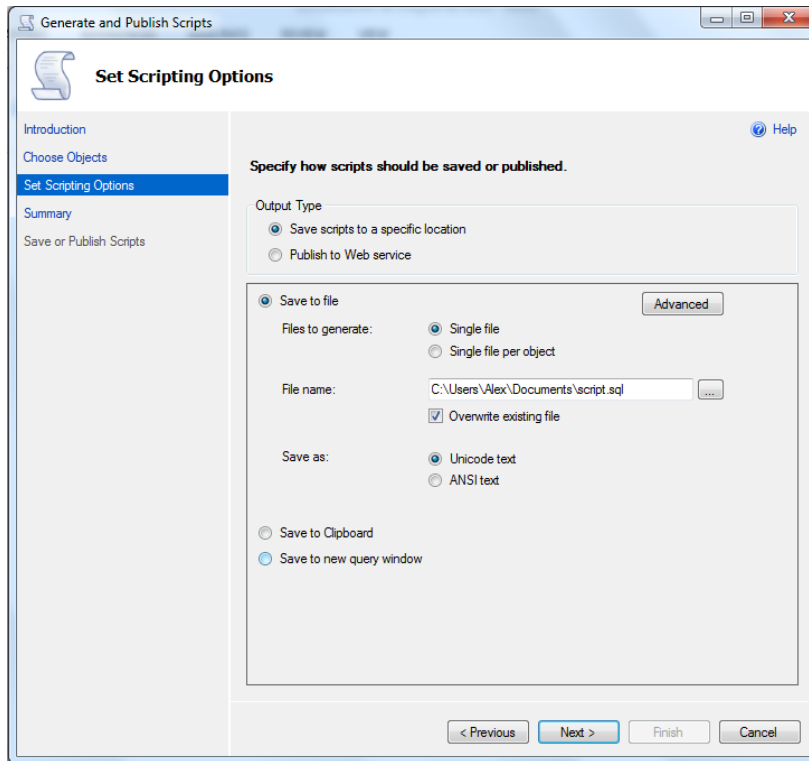
Captura 1



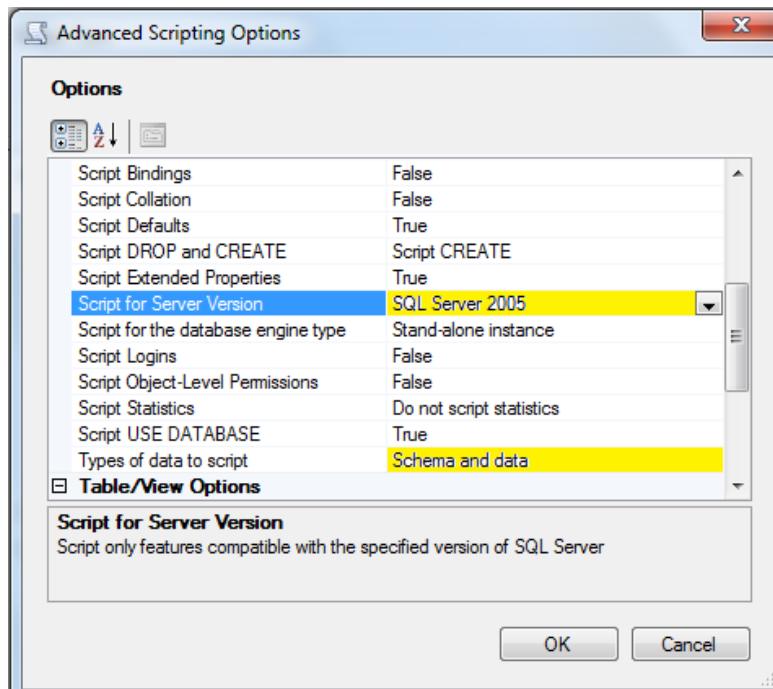
Captura 2



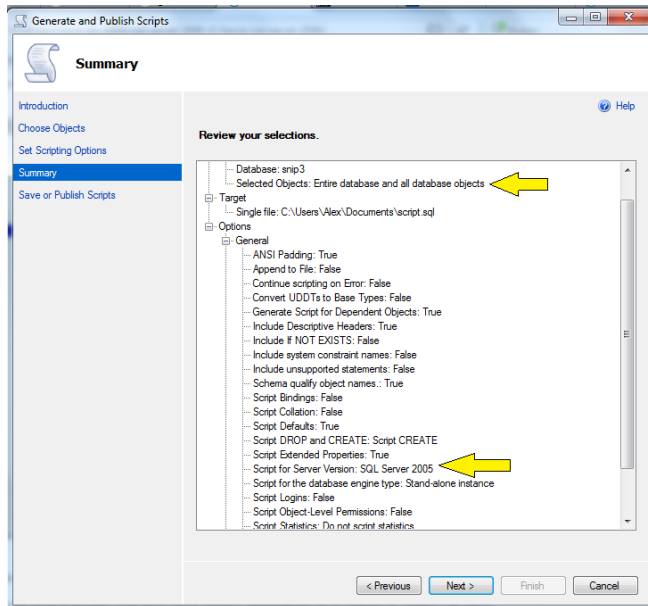
Captura 3



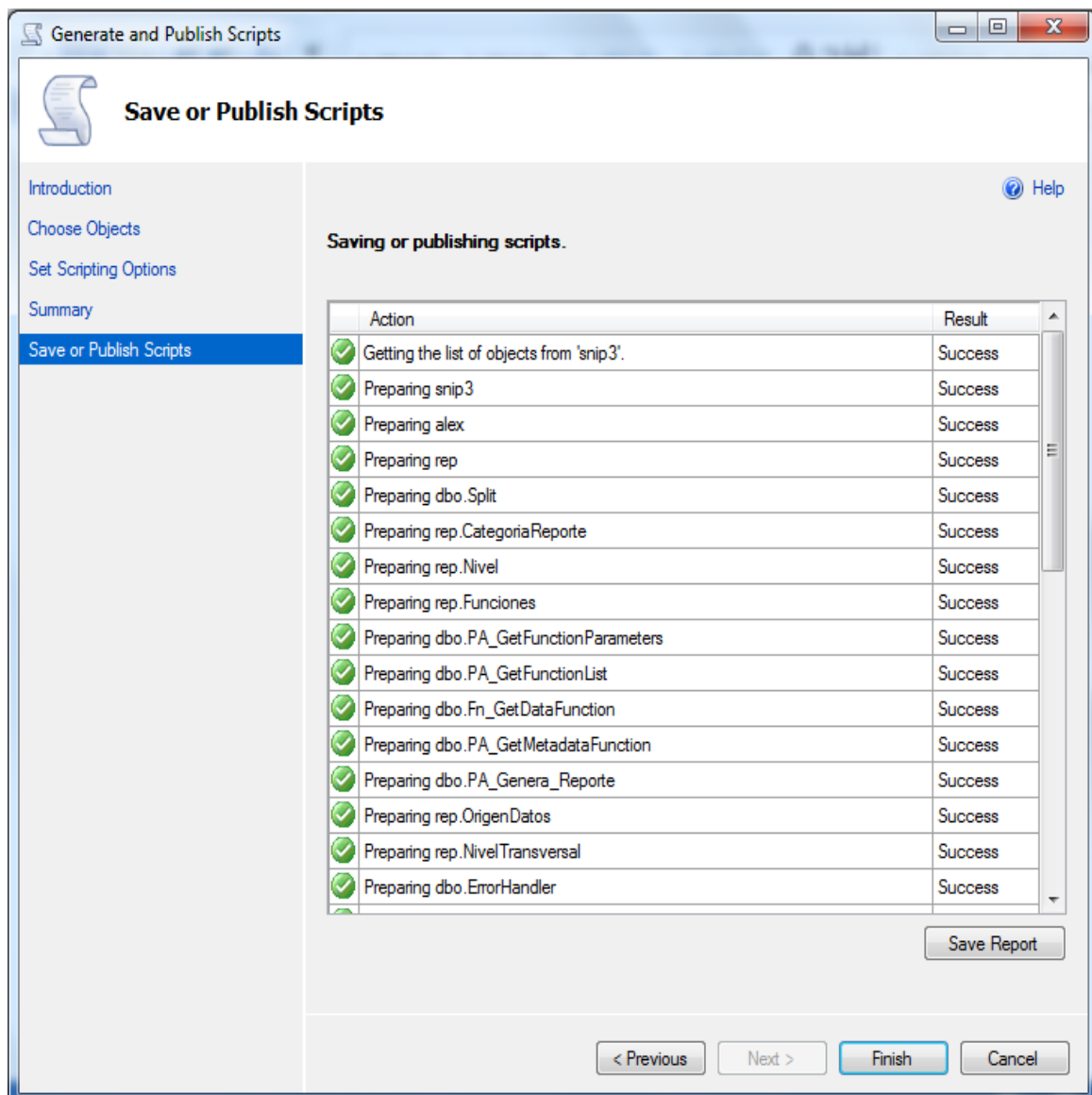
Captura 4



Captura 5



Captura 6



Captura 7